Item response theory based on deep learning with independent student and item networks

Emiko Tsutsumi

Graduate School of Informatics and Engineering
The University of Electro-Communications

March 2023

Supervisory Committee:

Prof. Maomi Ueno

Prof. Yusaku Yamamoto

Prof. Hayaru Shouno

Prof. Akihiro Kashihara

Assoc. Prof. Masaki Uto

論文の和文概要

論文題目	Item response theory based on deep learning with independent student and item networks
氏　　名	堤　瑛美子

近年，教育現場ではオンラインラーニングシステムで収集された教育ビッグデータをいかに有効に活用するかが課題となっている．人工知能分野では，機械学習を用いて学習者の課題への反応を予測することにより，学習者への適切な支援を行うアダプティブラーニングが注目されている．本研究では深層学習手法と項目反応理論を組み合わせ，パラメータの解釈性をもちながら高精度な反応予測を可能とする新たな項目反応理論を提案する．提案手法は学習者の項目への反応を二つの独立な学習者ネットワークと項目ネットワークで表現し，項目特性に依存せずに能力値を推定することができる．評価実験では，提案手法が最先端の反応予測手法と同程度の予測精度を示し，解釈性の高いモデルである従来の項目反応理論を上回る能力推定精度を示した．

Abstract

Knowledge tracing (KT), the task of tracking the knowledge state of a student over time, has been assessed actively by artificial intelligence researchers. Recent reports have described that Deep-IRT, which combines Item Response Theory (IRT) with a deep learning method, provides superior performance. It can express the abilities of each student and the difficulty of each item as in IRT. Nevertheless, its interpretability is inadequate compared to that of IRT because the ability parameter depends on each item. Deep-IRT implicitly assumes that items with the same skills are equivalent, which does not hold when item difficulties for the same skills differ greatly. For identical skills, items that are not equivalent hinder the interpretation of a student's ability estimate. To overcome those difficulties, this study proposes a new IRT based on deep learning that models a student response to an item using two independent networks: a student network and an item network. The proposed method learns student parameters and item parameters independently to avoid impairing the predictive accuracy. Moreover, we propose a novel hypernetwork architecture for the proposed method to balance both the current and the past data in the latent variable storing a student's knowledge states. Results of experiments demonstrate that the proposed method improves both the prediction accuracy and the interpretability of earlier KT methods.

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation comprehensively describes the research content of my doctoral studies at University of Electro-Communications, Tokyo, Japan. I am grateful to a large number of people who have helped me to accomplish my work. First of all, I would like to express my sincere gratitude to my supervisor, Professor Maomi Ueno, for his valuable comments, suggestions, and encouragement throughout the research period. Secondly, I would like to acknowledge Professor Yusaku Yamamoto, Professor Hayaru Shouno, Professor Akihiro Kashihara, and Associate Professor Masaki Uto. Their comments and suggestions on my research presentations were very helpful in improving my research and completing this thesis. Thirdly, I would also like to thank Professor Shuiti Kawano and Associate Professor Yu Nishiyama for their valuable advice and various help. This research was supported by JSPS KAKENHI Grant Numbers JP19H05663, JP22K19825, and JP22J15279. Finally, I would like to express my gratitude to my parents for their tremendous encouragement.

Emiko Tsutsumi
March 2023

1 Introduction

Recently, along with the advancement of online education, Knowledge Tracing (KT) has attracted broad attention for helping students to learn effectively by presenting optimal problems and a teacher's support [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Important tasks of KT are tracing the student's evolving knowledge state and discovering concepts that the student has not mastered based on the student's prior learning history data. Accurately predicting a student's performance (correct or incorrect responses to an unknown item) is important for adaptive learning.

Many researchers have developed various methods to solve KT tasks. Genereally speaking, KT methods are divisible into probabilistic approaches and deep-learning approaches. Bayesian Knowledge Tracing (BKT), a traditional and well known probabilistic model for KT [1], employs a Hidden Markov Model to trace a process of student ability growth. BKT estimates whether the student has mastered the skill or not according to the student's past response data. It then predicts the student's responses to unknown items. Researchers have proposed several BKT variants to improve interpretability [2, 3, 4, 5, 14, 15, 16, 17, 18]. The BKT models predict a student's knowledge state using only simple discrete values. Therefore, they are inflexible with the student knowledge state changes. Moreover, they assume a single dimension of the ability. They are unable to capture the multi-dimensional ability sufficiently or predict performance precisely. Recently, Item Response Theory (IRT) [19], which is used in the test theory area, has come to be used for KT [20, 21]. Actually, IRT predicts a student's correct answer probability to an item based on the student's latent ability parameter and item characteristic parameters. Several studies have extended standard IRT models to ascertain student ability changes for learning processes with the Hidden Markov process [20, 21, 22, 23, 24, 25, 26, 27]. These are regarded as generalized models of BKT and IRT because they estimate the ability as a continuous hidden variable following a Hidden Markov process.

Actually, a learning task is associated with multiple skills. Students must master the knowledge of multiple skills to solve a task. However, BKT and IRT have a restriction: they express only uni-dimensional ability. Therefore, BKT and IRT are unable to capture the multi-dimensional ability sufficiently. They are unable to predict the performance precisely.

To overcome the limitations, Deep Knowledge Tracing (DKT) [6] was proposed as the first deep-learning-based method. To predict student performance, DKT employs Long short - term memory (LSTM) [28]. LSTM relaxes the restrictions of skill separation and binary state assumptions. That earlier study shows that DKT

can predict a student's performance more precisely than traditional models, such as BKT, can. However, the hidden states include a summary of the past sequence of learning history data in LSTM. Therefore, DKT does not explicitly treat the student's ability of each skill.

To improve the DKT performance, various deep-learning-based methods have been proposed [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. Especially, the dynamic key-value memory network (DKVMN) was developed to exploit the relations among underlying skills and to trace the respective knowledge states using a Memory-Augmented Neural Network and attention mechanisms [8]. It can estimate the relations between underlying skills and items addressed by students. In addition, DKVMN has a memory updating component to allow forgetting and updating of the latent variable memory, which stores the students' knowledge states during the learning process [8]. Furthermore, to improve the explanatory capabilities of the parameters, Deep-IRT was proposed by combining DKVMN with an IRT module [7]. In fact, Deep-IRT can estimate a student's ability and an item's difficulty just as standard IRT models can. However, the ability parameter of the Deep-IRT depends on each item characteristic because it implicitly assumes that items with the same skills are equivalent. The assumption does not hold when the item difficulties for the same skills differ greatly. Items for the same skills which are not equivalent hinder interpretation of estimates of a student's ability.

The self-attentive knowledge tracing (SAKT) method [40] is the first method to employ an attention mechanism: the Transformer method [41], for KT. To predict student performance, SAKT identifies the relation between skills and an item addressed by a student from past learning data. Most recently, Gosh et al. (2020) proposed attentive knowledge tracing (AKT) [12], which incorporates a forgetting function of past data to attention mechanisms. AKT optimizes the parameters to weight the data necessary for student performance prediction from past learning data. Additionally, they pointed out a shortcoming by which earlier KT methods assumed that items with identical skills are equivalent. To overcome that shortcoming, they employed both items and skills as inputs. AKT provides state-of-the-art performance of future learner performance prediction. However, the interpretability of the parameters remains inadequate because AKT cannot express a student's ability transition of each skill.

Earlier studies have tackled the development of deep-learning-based methods to give parameter interpretability similarly to IRT models, but those studies have not achieved it for student ability parameters, which are most important for student modeling. An important shortcoming is the difficulty of incorporating the ability

parameters and item parameters independently into deep-learning-based methods so as not to degrade prediction accuracy. This study addresses that shortcoming.

Recent studies of deep learning have demonstrated that redundancy of parameters for training data reduces generalization error, contrary to Occam's razor. The studies also clarify the reasons underlying that finding [42, 43, 44]. Based on reports of state-of-the-art studies, this study proposes novel IRTs based on deep learning that model a student's response to an item by two independent redundant networks: a student network and an item network [13, 45]. The proposed method learns student parameters and item parameters independently to avoid impairment of the predictive accuracy. Therefore, the ability parameters of the proposed method are independent of each item's characteristics.

First, we propose a new IRT based on deep learning for test theory which has two independent redundant networks assuming that the ability is constant throughout the learning process [46]. This method is a new method of assessing learner competence values that solves the IRT problem. Evaluating the abilities of numerous students on a single scale requires linkage of students' abilities estimated from different tests [47, 48, 49, 50]. Although linkage techniques of IRT assume random sampling of students' abilities from a standard normal distribution, students' abilities have no guarantee of being sampled randomly from a standard normal distribution. On the other hand, the proposed method can express actual students' abilities distributions flexibly because it does not follow a standard normal distribution. Therefore, it estimates students' abilities with high accuracy when the students are not sampled randomly from a single distribution or when there are no common items among the different tests. The two independent networks provide a more reliable and robust ability estimation for actual data than IRT does.

Next, we propose a new IRT based on deep learning for knowledge tracing that estimates dynamic changes of student abilities in the learning process and predicts student performances [13]. This proposed method employs memory network architecture to reflect dynamic changes of student abilities as DKVMN does.The memory updating component in DKVMN is more effective than the forgetting function of AKT because it updates the current latent variable, which stores the students' skills and abilities using only the immediately preceding values. Because the estimated ability parameters are independent of each item's characteristics, they have higher interpretability than those of earlier Deep-IRT [7].

However, room for improvement remains in the prediction accuracy of the proposed method. In fact, the forgetting parameters which control the degree of forgetting the past latent variable are optimized from only the current input data: The

student's latest response to an item. It might degrade the prediction accuracy of the proposed method because the latent variable only insufficiently reflects the past data. As a result, it might interrupt the accurate estimation of the ability transition in a long learning process. It should use not only the current input data but also past latent variables to optimize the forgetting parameters. However, when learning the proposed method using both current and past input data, it is difficult to optimize the weight parameters directly because the number of parameters increases dynamically. To resolve that difficulty, we combine a novel hypernetwork with the proposed method because it optimizes the degree of forgetting of the past latent variables and thereby avoids greatly increasing the number of parameters.

Recent studies in the field of Natural Language Processing (NLP) have proposed several hypernetworks to optimize the latent variables and the weights of the hidden layers for LSTM [51, 52]. Some hypernetworks scale the latent variables and columns of all weight matrices expressing a context-dependent transition [51, 52, 53, 54, 55, 56, 57, 58]. No report of the relevant literature has described a study of the use of hypernetworks for KT methods. Using the proposed method, the proposed hypernetwork balances both current input data and past latent variables that store a student's knowledge state in the learning process. Before the model updates the latent variable, it optimizes not only the weights of the forgetting parameters but also the past latent variables in the hypernetwork. Although Tsutsumi et al. [45] proposed a hypernetwork for KT, they did not describe any related details: only the conceptual idea. Tsutsumi et al. [13, 45] improved the parameter interpretability. However, their prediction accuracies did not outperform AKT, which provided the best prediction performance among the earlier methods. In contrast, this study proposes a novel hypernetwork architecture to optimize the balance between the latest input data and the past latent variables.

As mentioned before, the original tasks of KT are discovering concepts that the student has not mastered and presenting optimal items by predicting the student's responses to unknown items [1, 5, 6, 7, 8]. On the other hand, for adaptive learning or adaptive testing, online learning systems with IRT estimate the student ability and the item parameters and presents optimal problems such that the students' correct probability is 0.5 [59, 60]. The proposed method can realize not only KT but also the online learning systems with IRT.

We conducted experiments to compare the proposed method's performance and those of earlier KT methods. Surprisingly, the results demonstrate that the proposed method improves the prediction accuracy and the interpretability of earlier KT methods including AKT, although the parameters of the proposed method are far

more numerous than those used for earlier methods.

2 Related Works

2.1 Item Response Theory

There are many item response theory (IRT) models [19, 48, 61]. This subsection briefly introduces two-parameter logistic model (2PLM): an extremely popular IRT model. For 2PLM, u_{ij} represents the response of student i to item j $(1..., J)$ as

$$u_{ij} = \begin{cases} 1 & (\textit{student } i \textit{ answers correctly to item } j), \\ 0 & (\textit{otherwise}). \end{cases}$$

In 2PLM, the probability of a correct answer given to item j by student i with ability parameter $\theta_i \in (-\infty, \infty)$ is assumed as

$$\begin{aligned} P_j(\theta_i) &= P(u_{ij} = 1 \mid \theta_i) \\ &= \frac{1}{1 + \exp(-1.7a_j(\theta_i - b_j))}, \end{aligned} \tag{1}$$

where $a_j \in (0, \infty)$ represents the j-th item's discrimination parameter expressing the discriminatory power for student's abilities, and $b_j \in (-\infty, \infty)$ is the j-th item's difficulty parameter representing the degree of difficulty.

From Bayes' theorem, the posterior distribution of an ability parameter $g(\theta|\boldsymbol{u})$ is given as

$$g(\theta|\boldsymbol{u}) = \frac{L(\theta|\boldsymbol{u})f(\theta)}{h(\boldsymbol{u})}, \tag{2}$$

where $L(\theta|\boldsymbol{u})$ is a likelihood, $f(\theta)$ is a prior distribution, and $h(\boldsymbol{u})$ is a marginal distribution:

$$h(\boldsymbol{u}) = \int_{-\infty}^{\infty} L(\theta|\boldsymbol{u})f(\theta)d\theta. \tag{3}$$

The parameters are estimated using the expected a priori (EAP) method, which is known to maximize the prediction accuracy theoretically as

$$\hat{\theta} = \int_{-\infty}^{\infty} \theta g(\theta|\boldsymbol{u})d\theta. \tag{4}$$

Because calculating the parameters analytically is difficult, numerical calculation methods such as Markov Chain Monte Carlo methods (MCMC) are generally used.

Actually, IRT models are known to have high interpretability. However, in standard IRT models, the ability is assumed to be constant throughout the learning process. Therefore, a student's ability changes are not reflected in the models. Recently, several studies have extended standard IRT models to capture student's ability changes for the learning processes with the Hidden Markov process[20, 21, 22, 23, 24, 25, 26]. These are regarded as generalized models of BKT and IRT because they estimate the ability as a continuous hidden variable following a Hidden Markov process.

For example, Temporal IRT (TIRT) is a Hidden Markov IRT with a parameter to forget past response data [21]. In TIRT, the probability of a correct answer assigned to item j by student i at time t with ability parameter θ_{it} is assumed as

$$P_{ij}(x_{ij} = 1 \mid \theta_{it}) = \frac{1}{1 + \exp\left(-\tilde{a}_{\Delta_t}(\theta_{it} - b_j)\right)}, \tag{5}$$

$$\tilde{a}_{\Delta_t} = \frac{a_j}{\sqrt{1 + \epsilon a_j^2 \Delta_t}}, \tag{6}$$

where Δ_t is a difference between the current time t and the past time t_j when the student answered to item j. $a_j \in (0, \infty)$ is the j-th item's discrimination parameter at time t. In addition, $b_j \in (-\infty, \infty)$ is the j-th item's difficulty parameter representing the degree of difficulty. Furthermore, $\theta_{it} \in (-\infty, \infty)$ represents the student i ability at time t. The prior of θ_{it} is a normal distribution described as $\theta_{i0} \sim \mathcal{N}(0, 1)$, $\theta_{it} \sim \mathcal{N}(\theta_{it-1}, \epsilon)$. The parameters are estimated using MCMC method as in 2PL IRT. Moreover, ϵ is a variance of θ_{it} and a forgetting parameter (tuning parameter), which determines the forgetting degree of the past data. The smoothness of a student's ability transition can be controlled by ϵ. Therefore, as ϵ increases, the fluctuation range of the true ability increases at each time point.

However, these IRT models incorporate the assumption of a single dimension of the ability. In other words, they consider completely independent multiple skills. Apparently, these are unable to accommodate items that require different skills.

2.2 Deep Knowledge Tracing

Deep knowledge tracing (DKT) [6] was proposed as the first deep-learning-based method. It exploits recurrent neural networks and Long short - term memory (LSTM) [28] to simulate transitions of ability. It can capture complex multidimensional features of both items and students and can relax the limitations of traditional methods such as independence between skills. An earlier study demonstrated that DKT outperformed BKT [1] in terms of predictive accuracy [6]. However, DKT

summarizes a student's ability of all skills in one hidden state, which makes it difficult to trace the degree to which a student has mastered a certain skill and to pinpoint concepts with which a student is proficient or unfamiliar.

2.3 Dynamic Key-value Memory Network

To improve the DKT interpretability, researchers have undertaken great efforts to propose novel methods for use with KT [29, 30, 31, 32, 34, 35, 36, 37, 38, 39, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71]. Specifically, a dynamic key-value memory network (DKVMN) exploits a memory-augmented neural network along with attention mechanisms to trace student abilities in different dimensions [8]. Figure 1 presents a brief illustration.

The salient feature of DKVMN is that it assumes N underlying skills and relations among the input (skills). Underlying skills are stored in key memory $\boldsymbol{M}^k \in \mathbb{R}^{N \times d_k}$. Value memory $\boldsymbol{M}_t^v \in \mathbb{R}^{N \times d_v}$ holds abilities of underlying skills at time t. Here, d_k and d_v are tuning parameters. To express the skill of j-th item, the input of DKVMN is an embedding vector $\boldsymbol{s}_j \in \mathbb{R}^{d_k}$ of skill tag of item j. DKVMN predicts the performance of item j at time t as explained below.

First, DKVMN calculates the attention, which indicates how strongly an item j is related to each skill as

$$w_{jl} = \text{Softmax}\left(\boldsymbol{M}_l^k \boldsymbol{s}_j\right), \tag{7}$$

where M_l^k represents a l th row vector, and w_{jl} signifies the degree of strength of the relation between the latent skill l and the skill of item j addressed by a student at time t. Also, $\text{Softmax}(z_i) = \exp(z_i)/\sum_j \exp(z_j)$ and is differentiable. Next, student vector $\boldsymbol{\theta}_1^{(t,j)}$ is calculated using the weighted sum of value memory.

$$\boldsymbol{\theta}_1^{(t,j)} = \sum_{l=1}^{N} w_{jl} \left(\boldsymbol{M}_{tl}^v\right)^\top, \tag{8}$$

where $\boldsymbol{M}_{tl}^v$ represents the l-th row vector. Finally, it concatenates $\boldsymbol{\theta}_1^{(t,j)}$ with $\boldsymbol{s}_j$ and predicts a correct probability P_{jt} for an item j as

$$\boldsymbol{\theta}_2^{(t,j)} = \tanh\left(\boldsymbol{W}^{(\theta_2)}\left[\boldsymbol{\theta}_1^{(t,j)}, \boldsymbol{s}_j\right] + \boldsymbol{\tau}^{(\theta_2)}\right), \tag{9}$$

$$P_{jt} = \sigma\left(\boldsymbol{W}^{(P_{jt})}\boldsymbol{\theta}_2^{(t,j)} + \boldsymbol{\tau}^{(P_{jt})}\right), \tag{10}$$

where $[\cdot]$ denotes a concatenation of vectors and $\sigma(\cdot)$ represents the sigmoid function defined by $\sigma(z) = 1/(1+\exp(-z))$. In this thesis, we express $\boldsymbol{W}^{(\cdot)}$ as the weight matrix and weight vector, and $\boldsymbol{\tau}^{(\cdot)}$ as the bias vector and scalar. Reportedly, DKVMN

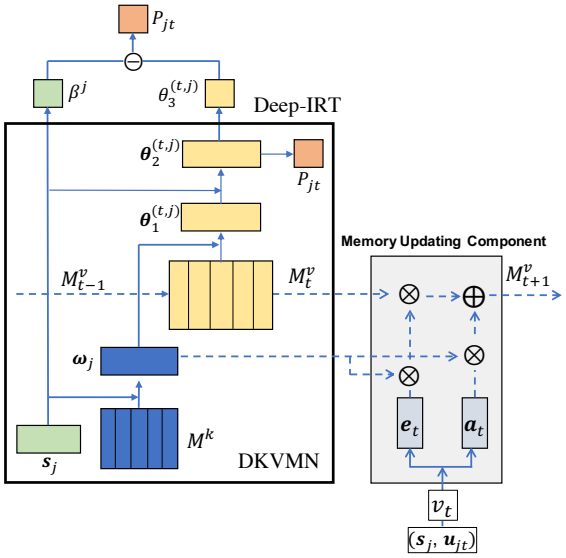


Figure 1: Network architecture of DKVMN and Deep-IRT. The underside of the structure describes DKVMN; the whole structure describes Deep-IRT. The blue components represent the process of getting the attention weight. The yellow components are associated with the student network and the process of updating the value memory. The green components are associated with the item network. The designation $\ominus$ represents subtraction.

has the capability of predicting performance accurately. However, unfortunately, it lacks interpretability of the parameters.

2.4 Deep-IRT

To improve the DKVMN interpretability, Deep-IRT is implemented by combining DKVMN with an IRT module [7]. Deep-IRT exploits both the strong prediction ability of DKVMN and the interpretable parameters of IRT. Fig.1 presents a simple illustration.

Deep-IRT adds a hidden layer to DKVMN to gain applicable ability and item difficulty. Specifically, when a student attempts item j at time t, an ability $\theta_3^{(t,j)}$ and item difficulty β^j are calculated as described below.

$$\theta_3^{(t,j)} = \tanh\left(\boldsymbol{W}^{(\theta_3)}\boldsymbol{\theta}_2^{(t,j)} + \boldsymbol{\tau}^{(\theta_3)}\right), \tag{11}$$

$$\beta^j = \tanh\left(\boldsymbol{W}^{(\beta)}\boldsymbol{s}_j + \boldsymbol{\tau}^{(\beta)}\right), \tag{12}$$

The prediction is based on the difference between $\theta_3^{(t,j)}$ and β^j such as IRT.

$$P_{jt} = \sigma\left(3.0 * \theta_3^{(t,j)} - \beta^j\right). \tag{13}$$

Here, ability $\boldsymbol{\theta}_2^{(t,j)}$ is calculated using $\boldsymbol{s}_j$ in equation (9), which depends on the item to solve because it implicitly assumes that items with the same skills are equivalent. In other words, this method cannot reflect the characteristics of each item. In fact, the ability estimate for the same student and time might differ if the student attempts a different item. An important difficulty is that a student's ability, which depends on each item, hinders the interpretability of the parameters.

2.5 Attentive Knowledge Tracing

Gosh et al. (2020) proposed attentive knowledge tracing (AKT) [12], which combines the attention-based model with the Rasch model, which is also known as the 1PLM IRT model [72]. It is noteworthy that AKT incorporates a forgetting function for past data into attention-based neural networks. Attention weights in AKT express the relation between a student's latest data and past data, decaying exponentially during the learning process. Specifically, AKT calculates the attention weight matrix $\boldsymbol{\alpha}$ as

$$\alpha_{t,\lambda} = \frac{\exp\left(f_{t,\lambda}\right)}{\sum_{\lambda'} \exp\left(f_{t,\lambda'}\right)}, \tag{14}$$

$$f_{t,\lambda} = \frac{\exp\left(-\eta d(t,\lambda)\right) \cdot \boldsymbol{q_t}^\top \boldsymbol{k}_\lambda}{\sqrt{D_k}}, \tag{15}$$

where $\eta > 0$ is a decay rate parameter and $d(t,\lambda)$ is a temporal distance measure between time steps t and λ. In addition, $\boldsymbol{q_t} \in \mathbb{R}^{D_k}$ denotes the query corresponding to items to which the student responds at time 1 to t, $\boldsymbol{k_\lambda} \in \mathbb{R}^{D_k}$ denotes the key for the item at time step λ and D_k denotes dimensions of the key matrix [12]. The attention weight $\boldsymbol{\alpha}$ decays as the distance between the current input time and the past input time increases. Furthermore, $d(t,\lambda)$ with $\lambda \leq t$ is obtained as explained below.

$$d(t,\lambda) = |t-\lambda| \sum_{t'=\lambda+1}^{t} \frac{\frac{\boldsymbol{q_t}^\top \boldsymbol{k}_{t'}}{\sqrt{D_k}}}{\sum_{1 \leq \lambda' \leq t'} \frac{\boldsymbol{q_t}^\top \boldsymbol{k}_{\lambda'}}{\sqrt{D_k}}}, \quad \forall t' \leq t. \tag{16}$$

In fact, $d(t,\lambda)$ adjusts the distance between consecutive time indices according to how the past input is related to the current input [12].

Additionally, they pointed out that the earlier KT methods assumed that items with the same skills are equivalent. To resolve the difficulty, AKT employs both items and skill inputs. Results show that, among the earlier KT methods, AKT provides the best performance for predicting the students' responses. Nevertheless, the interpretability of its parameters remains inadequate because it cannot express a student's ability transition for each skill.

3 IRT Based on Deep Learning with Independent Student and Item Networks

Earlier studies have tackled developing deep-learning-based methods to give parameter interpretability similar to IRT models, but those studies have not achieved it for student ability parameters, which are most important for student modeling. The problem is the difficulty of independently incorporating the ability parameters and item parameters into deep-learning-based methods so as not to degrade prediction accuracy. This study addresses that problem.

3.1 IRT Based on Deep Learning for Test Theory

To improve the parameter interpretability of deep learning method, first, we propose a novel item response theory based on deep learning for test theory. The proposed method estimates parameters using two independent networks: a student network and an item network. However, in general, independent networks are known to have less prediction accuracy than dependent networks have. Recent studies of deep learning have demonstrated that redundancy of parameters (deep layers of hidden variables) reduces generalization error, contrary to Occam's razor [42, 43, 44]. Based on state-of-the-art reports, The proposed method constructs two independent redundant deep networks as presented in Figure 2.

Furthermore, the proposed method solves the IRT problem for E-testing. E-testing involves the delivery of examinations and assessments on screen, using either local systems or web-based systems. In general, e-testing provides automatic assemblies of uniform test forms, for which each form comprises a different set of items but which still has equivalent measurement accuracy [73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83]. Uniform test forms are assembled for which all forms have equivalent qualities for equal evaluation of examinees who have taken different test forms. Students' test scores should be guaranteed to become equivalent, even if different students with the same ability take different tests. However, because it is difficult to develop perfectly uniform test forms, the calibration process is fundamentally important when multiple test forms are used.

To resolve this difficulty, IRT has been used as a calibration method. Evaluating the abilities of numerous students on a single scale requires linkage of students' abilities estimated from different tests [47, 48, 49, 50]. Although linkage techniques of IRT assume random sampling of students' abilities from a standard normal distribution, students' abilities have no guarantee of being sampled randomly from a

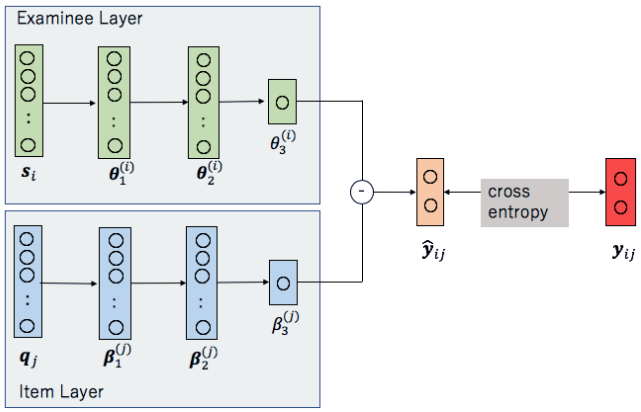


Figure 2: Network architecture of the proposed method with independent student and item networks for test theory. The green components are associated with the student network. The blue components are associated with the item network. The designations $\ominus$ represents element-wise multiplication and subtraction.

standard normal distribution. In addition, IRT requires the assumption of local independence between items. Although it estimates students' abilities on the assumption that a student's response to each item is independent, items might not always have local independence. However, the proposed method can express actual students' abilities distributions flexibly because the method does not assume a standard normal distribution of the abilities. It estimates students' abilities with high accuracy when they are not sampled randomly from a single distribution or when there are no common items among the different tests. Furthermore, the proposed method estimates students' abilities while including consideration of the relations among items. Therefore, it provides a more reliable and robust ability estimation for actual data than IRT does. It is expected to have highly interpretable parameters without impairment of the estimation accuracy.

3.1.1 Student Network

To express the i-th student, the encode of student network is a one-hot vector $\boldsymbol{s}_i \in \{0,1\}^I$, where I represents the number of students. The i-th element is 1; the other elements are 0s. The student network comprises three layers as described below.

$$\boldsymbol{\theta}_1^{(i)} = \tanh\left(\boldsymbol{W}^{(\theta_1)}\boldsymbol{s}_i + \boldsymbol{\tau}^{(\theta_1)}\right), \tag{17}$$

$$\boldsymbol{\theta}_2^{(i)} = \tanh\left(\boldsymbol{W}^{(\theta_2)}\boldsymbol{\theta}_1^{(i)} + \boldsymbol{\tau}^{(\theta_2)}\right), \tag{18}$$

$$\theta_3^{(i)} = \boldsymbol{W}^{(\theta_3)}\boldsymbol{\theta}_2^{(i)} + \tau^{(\theta_3)}. \tag{19}$$

Here, $\boldsymbol{W}^{(\theta_1)}$ and $\boldsymbol{W}^{(\theta_2)}$ are the weight matrices given as

$$\boldsymbol{W}^{(\theta_1)} = \begin{pmatrix} w_{11}^{(\theta_1)} & w_{12}^{(\theta_1)} & \dots & w_{1I}^{(\theta_1)} \\ w_{21}^{(\theta_1)} & w_{22}^{(\theta_1)} & \dots & w_{2I}^{(\theta_1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|\boldsymbol{\theta}_1|1}^{(\theta_1)} & w_{|\boldsymbol{\theta}_1|2}^{(\theta_1)} & \dots & w_{|\boldsymbol{\theta}_1|I}^{(\theta_1)} \end{pmatrix},$$

$$\boldsymbol{W}^{(\theta_2)} = \begin{pmatrix} w_{11}^{(\theta_2)} & w_{12}^{(\theta_2)} & \dots & w_{1|\boldsymbol{\theta}_1|}^{(\theta_2)} \\ w_{21}^{(\theta_2)} & w_{22}^{(\theta_2)} & \dots & w_{2|\boldsymbol{\theta}_1|}^{(\theta_2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|\boldsymbol{\theta}_2|1}^{(\theta_2)} & w_{|\boldsymbol{\theta}_2|2}^{(\theta_2)} & \dots & w_{|\boldsymbol{\theta}_2||\boldsymbol{\theta}_1|}^{(\theta_2)} \end{pmatrix}.$$

Therein, $\boldsymbol{W}^{(\theta_3)}$ is the weight vector given as

$$\boldsymbol{W}^{(\theta_3)} = \begin{pmatrix} w_1^{(\theta_3)}, & w_2^{(\theta_3)}, & \dots, & w_{|\boldsymbol{\theta}_2|}^{(\theta_3)} \end{pmatrix}.$$

In addition, $\boldsymbol{\tau}^{(\theta_1)} = \left(\tau_1^{(\theta_1)}, \tau_2^{(\theta_1)}, ..., \tau_{|\boldsymbol{\theta}_1|}^{(\theta_1)}\right)^\top$ and $\boldsymbol{\tau}^{(\theta_2)} = \left(\tau_1^{(\theta_2)}, \tau_2^{(\theta_2)}, ..., \tau_{|\boldsymbol{\theta}_2|}^{(\theta_2)}\right)^\top$ are the bias parameters vectors; $\tau^{(\theta_3)}$ is the bias parameter. In this study, the last layer $\theta_3^{(i)}$ is expressed as the i-th student's ability parameter. Although the estimated student's ability parameters have discrimination, we standardize them according to the standard IRT models.

An overview of the calculation in terms of the student network is presented in Figure 3. Here, u_{ij} represents the response of student i to item j and the weight matrix $\boldsymbol{W}$ represents an estimate of the relation between a student's ability and all other students' abilities. Therefore, the proposed method does not require the assumption of random sampling students' abilities from a statistical distribution because it estimates a student's ability by adjusting the other students' ability estimates.

3.1.2 Item Network

Similarly, to express the j-th item, the encoding of the item network is a one-hot vector $\boldsymbol{q}_i \in \{0, 1\}^J$, where J stands for the number of items. The j-th element is 1; the other elements are 0s. The item network consists of three layers as follows.

$$\boldsymbol{\beta}_1^{(j)} = \tanh\left(\boldsymbol{W}^{(\beta_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(\beta_1)}\right), \tag{20}$$

$$\boldsymbol{\beta}_2^{(j)} = \tanh\left(\boldsymbol{W}^{(\beta_2)}\boldsymbol{\beta}_1^{(j)} + \boldsymbol{\tau}^{(\beta_2)}\right), \tag{21}$$

$$\beta_3^{(j)} = \boldsymbol{W}^{(\beta_3)}\boldsymbol{\beta}_2^{(j)} + \tau^{(\beta_3)}. \tag{22}$$

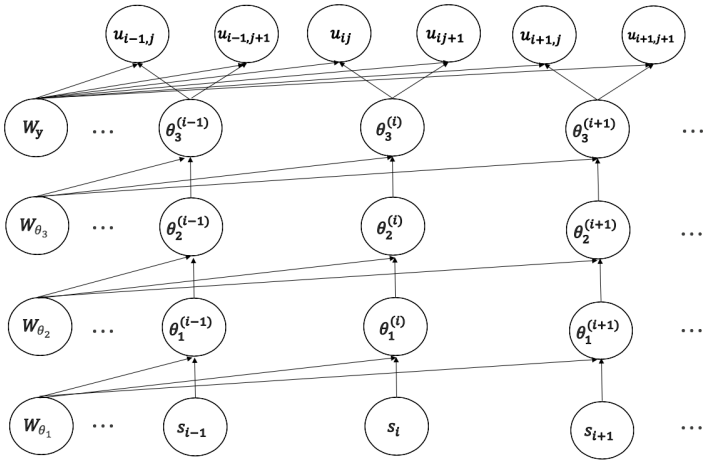


Figure 3: Student layer structure.

In addition, $\boldsymbol{W}^{(\beta_1)}$ and $\boldsymbol{W}^{(\beta_2)}$ are the weight matrices given as presented below.

$$\boldsymbol{W}^{(\beta_1)} = \begin{pmatrix} w_{11}^{(\beta_1)} & w_{12}^{(\beta_1)} & \dots & w_{1J}^{(\beta_1)} \\ w_{21}^{(\beta_1)} & w_{22}^{(\beta_1)} & \dots & w_{2J}^{(\beta_1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|\boldsymbol{\beta}_1|1}^{(\beta_1)} & w_{|\boldsymbol{\beta}_1|2}^{(\beta_1)} & \dots & w_{|\boldsymbol{\beta}_1|J}^{(\beta_1)} \end{pmatrix},$$

$$\boldsymbol{W}^{(\beta_2)} = \begin{pmatrix} w_{11}^{(\beta_2)} & w_{12}^{(\beta_2)} & \dots & w_{1|\boldsymbol{\beta}_1|}^{(\beta_2)} \\ w_{21}^{(\beta_2)} & w_{22}^{(\beta_2)} & \dots & w_{2|\boldsymbol{\beta}_1|}^{(\beta_2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|\boldsymbol{\beta}_2|1}^{(\beta_2)} & w_{|\boldsymbol{\beta}_2|2}^{(\beta_2)} & \dots & w_{|\boldsymbol{\beta}_2||\boldsymbol{\beta}_1|}^{(\beta_2)} \end{pmatrix}.$$

Here, $\boldsymbol{W}^{(\beta_3)}$ is the weight vector given as shown below.

$$\boldsymbol{W}^{(\beta_3)} = \begin{pmatrix} w_1^{(\beta_3)}, & w_2^{(\beta_3)}, & \dots, & w_{|\boldsymbol{\beta}_2|}^{(\beta_3)} \end{pmatrix}.$$

Additionally, $\boldsymbol{\tau}^{(\beta_1)} = \left(\tau_1^{(\beta_1)}, \tau_2^{(\beta_1)}, ..., \tau_{|\boldsymbol{\beta}_1|}^{(\beta_1)}\right)^\top$ and $\boldsymbol{\tau}^{(\beta_2)} = \left(\tau_1^{(\beta_2)}, \tau_2^{(\beta_2)}, ..., \tau_{|\boldsymbol{\beta}_2|}^{(\beta_2)}\right)^\top$ are the bias parameters vectors. $\tau^{(\beta_3)}$ is the bias parameter. For this study, we consider the last layer $\beta_3^{(j)}$ as the j th item's difficulty parameter. Similarly to sampling of students, this method does not assume random sampling of item difficulty parameters from a statistical distribution.

Then, the proposed method represents a student's correct response probability to an item using the difference between the student's ability parameter and the item difficulty parameter. Specifically, student i's correct response probability to j's item is described using a hidden layer $\boldsymbol{h}^{(i,j)} = (h_0^{(i,j)}, h_1^{(i,j)})^\top$ as

$$\boldsymbol{h}^{(i,j)} = (\boldsymbol{W}^{(y)})(\theta_3^{(i)} - \beta_3^{(j)}) + \boldsymbol{\tau}^{(y)}, \tag{23}$$

$$\begin{aligned}\hat{y}_{ij} &= \mathrm{Softmax}(\boldsymbol{h}^{(i,j)}) \\ &= \frac{\exp(h_1^{(i,j)})}{\exp(h_0^{(i,j)}) + \exp(h_1^{(i,j)})}.\end{aligned} \tag{24}$$

Here, $\boldsymbol{W}^{(y)} = (w_1^{(y)}, w_2^{(y)})^\top$ and $\boldsymbol{\tau}^{(y)} = (\tau_1^{(y)}, \tau_2^{(y)})^\top$ are the weight vector and bias parameters vector.

The proposed method does not assume random sampling of students' abilities and item difficulties from any statistical distribution. Instead, it uses a deep learning method to estimate the relation between a students' ability and all other students' abilities by maximizing the prediction accuracy of students' responses. The unique feature of this method is to estimate a student's ability by adjusting the other students' ability estimates.

3.1.3 Prediction of Student Response to an Item

In general, deep learning methods learn their parameters using the back-propagation algorithm by minimizing a loss function. The loss function of the proposed method employs cross-entropy, which reflects classification errors. It is calculated from the predicted responses $\hat{y}_{ij}$ and the true responses u_{ij} as

$$\ell(u_{ij}, \hat{y}_{ij}) = -u_{ij} \log \hat{y}_{ij} - (1 - u_{ij}) \log(1 - \hat{y}_{ij}). \tag{25}$$

Like other machine learning techniques, deep learning methods are biased to data they have encountered before. Therefore, the generalization capacity of the methods depends on the training data, which leads to sub-optimal performance. Consequently, the proposed method cannot predict responses of students or items accurately with an extremely small number of (in)correct answers. To overcome this shortcoming, cost-sensitive learning, which weights minority data over majority, has been used widely [84]. Therefore, we add the loss function based on a cost-sensitive

approach as

$$
\begin{aligned}
Loss_{class} = & \sum_{i}\sum_{j} \ell(u_{ij}, \hat{y}_{ij}) \\
& + \gamma_1 \sum_{i \in L_e} \sum_{j \in (u_{ij}=1)} \ell(u_{ij}, \hat{y}_{ij}) \\
& + \gamma_2 \sum_{i \in H_e} \sum_{j \in (u_{ij}=0)} \ell(u_{ij}, \hat{y}_{ij}) \\
& + \gamma_3 \sum_{j \in L_i} \sum_{i \in (u_{ij}=1)} \ell(u_{ij}, \hat{y}_{ij}) \\
& + \gamma_4 \sum_{j \in H_i} \sum_{i \in (u_{ij}=0)} \ell(u_{ij}, \hat{y}_{ij}),
\end{aligned} \tag{26}
$$

where L_e stands for a group of students whose correct answer rates are less than α_{L_e}, H_e denotes a group of students whose correct answer rates are more than α_{H_e}, L_i signifies a group of items of which correct answer rates are less than α_{L_i}, and H_i represents a group of items with correct answer rates that are more than α_{H_i}. Here, $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and $\alpha_{L_e}, \alpha_{H_e}, \alpha_{L_i}, \alpha_{H_i}$ are tuning parameters. All of the parameters are learned simultaneously using a popular optimization algorithm: adaptive moment estimation [85].

Although this method have high parameters interpretability, a student's ability is constant throughout a learning process. Therefore, it can not be applied to knowledge tracing.

3.2 IRT Based on Deep Learning for Knowledge Tracing

In this section, we propose a novel IRT based on deep learning for knowledge tracing which estimates dynamic changes of student abilities in the learning process and predicts student performances. The ability parameter of Deep-IRT [7] depends on each item because it implicitly assumes that items with the same skills are equivalent. That assumption does not hold when the item difficulties for the same skills differ greatly. Therefore, when the items for the same skills are not equivalent, it is difficult to interpret a student's ability estimate. To resolve the difficulty, this study proposes a novel IRT based on deep learning comprising two independent neural networks: the student network and the item deep network, as presented in Figure 4. The student network employs memory network architecture such as DKVMN to ascertain changes in student ability comprehensively. The item network includes inputs of two kinds: the item attempted by a student and the necessary skills to solve the item. Using outputs of both networks, the probability of a student answering

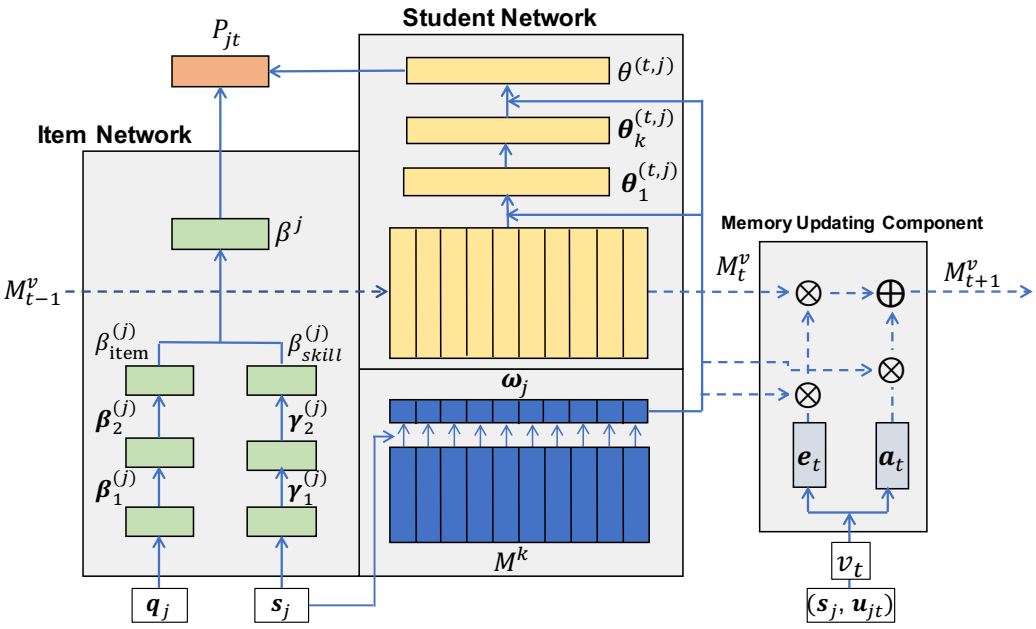


Figure 4: Network architecture of the proposed method with independent student and item networks for KT. The yellow components are associated with the student network. The green components are associated with the item network. In addition, the right side of figure presents the memory updating component. The designations $\otimes$ and $\oplus$ respectively represent element-wise multiplication and addition.

an item correctly can be calculated. The proposed method 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 more redundant than they are with earlier methods, based on state-of-the-art reports [42, 43, 44].

As mentioned before, the original tasks of KT are discovering concepts that the student has not mastered and presenting optimal items by predicting the student's responses to unknown items [1, 5, 6, 7, 8]. On the other hand, for adaptive learning or adaptive testing, online learning systems with IRT estimate the student ability and the item parameters and presents optimal problems such that the students' correct probability is 0.5 [59, 60]. The proposed method can realize not only KT but also the online learning systems with IRT.

The proposed method calculates P_{jt}, the probability of a correct answer assigned to the item j at time t, using the item difficulties and the student abilities, as shown hereinafter.

3.2.1 Item Network

In the item network, two difficulty parameters of item j are estimated: the item characteristic difficulty parameter β^j_{item} and the skill difficulty β^j_{skill}. The item characteristic difficulty parameter represents the unique difficulties of the item, excepting the required skill difficulty. The proposed method expresses item difficulty as the sum of the two difficulty parameters of β^j_{item} and β^j_{skill}.

In the proposed method, to express the j-th item, an input of the item network is an embedding vector $\boldsymbol{q}_j \in \mathbb{R}^{d_k}$ of item j. The item characteristic difficulty parameter of item j is calculated using a feed forward neural network as

$$\boldsymbol{\beta}^j_1 = \tanh\left(\boldsymbol{W}^{(\beta_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(\beta_1)}\right), \tag{27}$$

$$\boldsymbol{\beta}^j_{k'} = \tanh\left(\boldsymbol{W}^{(\beta_{k'})}\boldsymbol{\beta}^j_{k'-1} + \boldsymbol{\tau}^{(\beta_{k'})}\right), \tag{28}$$

$$\beta^j_{\text{item}} = \boldsymbol{W}^{(\beta_{\text{item}})}\boldsymbol{\beta}^j_k + \boldsymbol{\tau}^{(\beta_{\text{item}})}. \tag{29}$$

In this report, we represent $\{k \in \mathbb{N} | 2 \leq k' \leq k\}$ as the number of hidden layers determined depending on the prediction accuracy of actual data. The last layer β^j_{item} represents the j-th item characteristic difficulty parameter.

Similarly, to compute the difficulty of skills, the proposed method uses the input of necessary skills $\boldsymbol{s}_j \in \mathbb{R}^{d_k}$. The embedding vector $\boldsymbol{s}_j$ is calculated from the skill tag of item j.

$$\boldsymbol{\gamma}^j_1 = \tanh\left(\boldsymbol{W}^{(\gamma_1)}\boldsymbol{s}_j + \boldsymbol{\tau}^{(\gamma_1)}\right), \tag{30}$$

$$\boldsymbol{\gamma}^j_{k'} = \tanh\left(\boldsymbol{W}^{(\gamma_{k'})}\boldsymbol{\gamma}^j_{k'-1} + \boldsymbol{\tau}^{(\gamma_{k'})}\right), \tag{31}$$

$$\beta^j_{\text{skill}} = \boldsymbol{W}^{(\beta_{\text{skill}})}\boldsymbol{\gamma}^j_k + \boldsymbol{\tau}^{(\beta_{\text{skill}})}, \tag{32}$$

where $\{k \in \mathbb{N} | 2 \leq k' \leq k\}$. The last layer β^j_{skill} denotes the difficulty parameter of the required skills to solve the j-th item.

3.2.2 Student Network

In the student network, the proposed method calculates $\boldsymbol{\theta}^{(t,j)}_1$ based on the latent variable $\boldsymbol{M}^v_t$ expressing a student's latent knowledge state at time t, as

$$\boldsymbol{\theta}^{(t,j)}_1 = \sum_{l=1}^{N} w_{jl}\left(\boldsymbol{M}^v_{tl}\right)^\top, \tag{33}$$

where M^v_{tl} represents a l-th row vector and w_{tl} is the attention weight of underlying skill l. w_{tl} is estimated similarly to DKVMN in equation (7). Next, an interpretable

student's ability vector $\theta^{(t,j)}$ can be estimated as presented below.

$$\boldsymbol{\theta}_{k'}^{(t,j)} = \tanh\left(\boldsymbol{W}^{(\theta_{k'})}\boldsymbol{\theta}_{k'-1}^{(t,j)} + \boldsymbol{\tau}^{(\theta_{k'})}\right), \tag{34}$$

$$\theta^{(t,j)} = \sum_{l=1}^{N} w_{tl}\theta_{kl}^{(t,j)}, \tag{35}$$

where $\{k \in \mathbb{N} | 2 \leq k' \leq k\}$ and $\boldsymbol{\theta}_{k}^{(t,j)} = \{\theta_{k1}^{(t,j)}, \theta_{k2}^{(t,j)}, \cdots, \theta_{kN}^{(t,j)}\}$. Also, $\boldsymbol{\theta}_{k'}^{(t,j)} \in \mathbb{R}^{d_v}$ and $\boldsymbol{\theta}_{k}^{(t,j)} \in \mathbb{R}^{N}$. One important difference between the proposed method and Deep-IRT [7] is that the proposed method does not calculate $\boldsymbol{\theta}_{k}^{(t,j)}$ using features of items such as equations (6) and (8). Therefore, the ability parameter $\theta^{(t,j)}$ is independent of the difficulty parameters of the respective items. In addition, the value of $\boldsymbol{\theta}_{k}^{(t,j)}$ represents the abilities of the latent skills. In other words, $\boldsymbol{\theta}_{k}^{(t,j)}$ can be inferred as a measurement model, such as multidimensional IRT [86].

3.2.3 Prediction of Student Response to an Item

The proposed method calculates a student's response probability to an item using the difference between a student's ability $\theta^{(t,j)}$ to solve item j at time t and the sum of two difficulty parameters β_{item}^{j} and β_{skill}^{j}.

$$P_{jt} = \sigma\left(3.0 * \theta^{(t,j)} - (\beta_{\text{item}}^{j} + \beta_{\text{skill}}^{j})\right). \tag{36}$$

After the procedure, the latent value memory $\boldsymbol{M}_t^v$ is updated using the embedding vector of (s_j, u_{jt}) denoted as $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$ as in DKVMN [8]. Here, u_{jt} is the student's response to item j at time t: u_{jt} is 1 when the student answers the item correctly; it is 0 otherwise.

$$\boldsymbol{e}_t = \sigma(\boldsymbol{W}^e\boldsymbol{v}_t + \boldsymbol{\tau}^e), \tag{37}$$

$$\boldsymbol{a}_t = \tanh(\boldsymbol{W}^a\boldsymbol{v}_t + \boldsymbol{\tau}^a), \tag{38}$$

$$\tilde{\boldsymbol{M}}_{t+1,l}^v = \boldsymbol{M}_{t,l}^v \otimes (1 - w_{jl}\boldsymbol{e}_t)^\top, \tag{39}$$

$$\boldsymbol{M}_{t+1,l}^v = \tilde{\boldsymbol{M}}_{t+1,l}^v + w_{jl}\boldsymbol{a}_t^\top. \tag{40}$$

Therein, $\boldsymbol{W}^e \in \mathbb{R}^{d_v \times d_v}, \boldsymbol{W}^a \in \mathbb{R}^{d_v \times d_v}$ are weight matrices and $\boldsymbol{\tau}^e \in \mathbb{R}^{d_v}, \boldsymbol{\tau}^e \in \mathbb{R}^{d_v}$ are bias vectors. l is a underling skill and $\{l \in \mathbb{N} | 1 \leq l \leq N\}$. $\otimes$ represents the element-wise product. In equations (37) and (39), $\boldsymbol{e}_t$ controls how much the value memory forgets (remembers) the past ability. In addition, $\boldsymbol{a}_t$ in equations (38) and (40) controls how strongly current performance is reflected. It is noteworthy that $\boldsymbol{e}_t$ and $\boldsymbol{a}_t$, which control the degree of forgetting the past latent value memory $\boldsymbol{M}_t^v$, are optimized from only the student's latest response to an item u_{jt}.

In general, deep-learning-based methods learn their parameters using the back-propagation algorithm by minimizing a loss function. The loss function of the proposed method employs cross-entropy, which reflects classification errors. Then the cross-entropy of the predicted responses P_{jt} and the true responses u_{jt} is calculated as

$$\ell(u_{jt}, P_{jt}) = -\sum_{t} \left(u_{jt} \log P_{jt} + (1 - u_{jt}) \log(1 - P_{jt})\right). \tag{41}$$

All parameters are learned simultaneously using a well known optimization algorithm: adaptive moment estimation [85].

3.3 IRT Based on Deep Learning with Hypernetwork for Knowledge Tracing

The preceding section described the proposed method with independent student and item networks. However, room for improvement of the prediction accuracy remains because the parameters which control the degree of forgetting the past latent value memory $\boldsymbol{M}_t^v$ are optimized using only the student's latest response to an item. It might degrade the prediction accuracy of the proposed method because the latent value memory insufficiently reflects past data. As a result, it might present difficulty for accurate estimation of the ability transition in a long learning process. It should use not only the current input data but also past data to optimize the forgetting parameters. However, when using both current and past data, optimizing the weight parameters directly is difficult because the number of parameters increases dynamically.

Recent reports of studies conducted in the field of Natural Language Processing (NLP) have proposed extension components to LSTM [28] in the form of mutual gating of the current input data and previous hidden variable [51]. These extension components are called hypernetworks. In standard LSTM [28], the hidden variables change with time, but the weights used to update them are fixed values that are not optimized for each time point. To resolve this difficulty, various hypernetworks have been proposed to support the main recurrent neural network by optimizing the non-shared weights for each time point in the hidden layers [51, 53, 54, 55, 56, 57, 58]. Their results demonstrate that LSTM with a hypernetwork works better than the standard LSTM [28]. Furthermore, Melis et al. (2020) earlier proposed the "Mogrifier component," which is a kind of hypernetwork for LSTM in the field of NLP [53]. Mogrifier scales the hidden variables using not only the current inputs but also the output of the hidden variable at the earlier time point. They reported

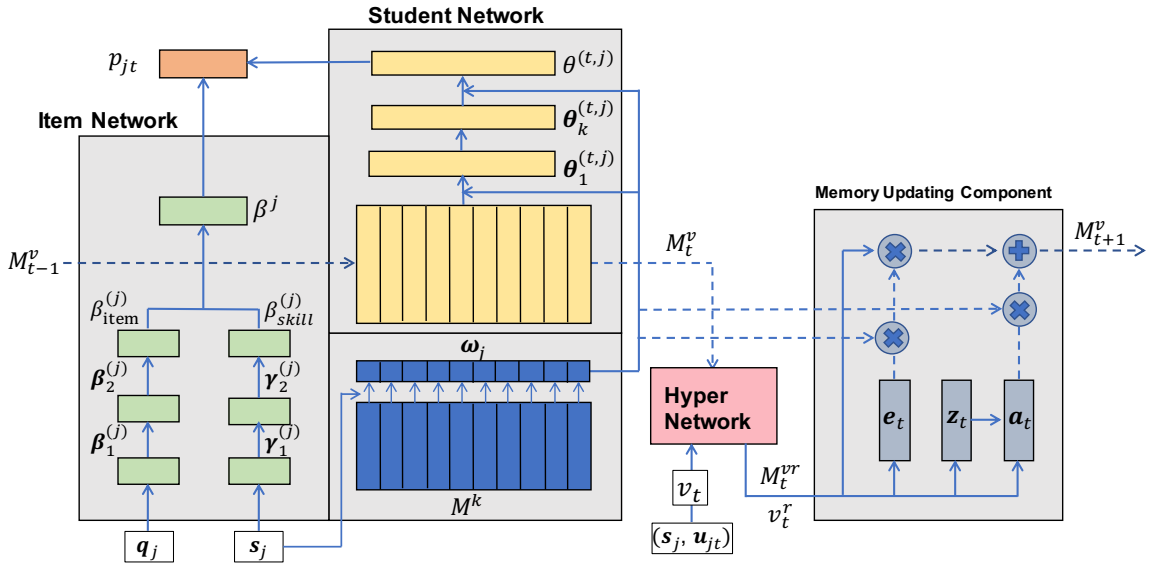


Figure 5: Memory updating component of the proposed method with hypernetwork. The proposed hypernetwork is located at the beginning of the Memory Updating Component. It estimates the optimal forgetting parameters by balancing both the current input data and the past latent variable before the model updates the latent variable.

that LSTM with the Mogrifier component outperforms the other methods for long input data lengths.

Inspired by the results obtained from those studies, we incorporate a novel hypernetwork into the memory updating component (in Figure 4), which updates the latent variable $\boldsymbol{M}^v_t$ expressing a student's knowledge state, to avoid greatly increasing the number of parameters. Although Tsutsumi et al. [45] proposed a hypernetwork for KT, that report presented no details but just its conceptual idea. This study proposes a novel hypernetwork architecture to optimize the balance between the latest input data and the past latent variables. No report of the relevant literature has described a study of the use of the hypernetworks for KT methods.

Figure 5 presents the proposed hypernetwork architecture and the memory updating component of the proposed method. The hypernetwork optimizes the degree of forgetting of past data in the proposed method and improves prediction accuracy with parameter interpretability. Specifically, before the method updates the latent variable $\boldsymbol{M}^v_{t+1}$, the proposed hypernetwork balances both the current input data $\boldsymbol{v}_t$ and the latent variable $\boldsymbol{M}^v_t$ using the past latent variables $\{\boldsymbol{M}^v_t, \boldsymbol{M}^v_{t-1}, \cdots, \boldsymbol{M}^v_{t-\lambda}\}$ at time $t-\lambda$ to t. Here, λ represents the degree of the past latent variables to be accessed. For the proposed method, we optimize λ for each learning dataset.

3.3.1 Hypernetwork

In the memory updating components of DKVMN and Deep-IRT [8, 7], the forgetting parameters are optimized only from current input data. Therefore, their value memory $\boldsymbol{M}_t^v$ might inadequately forget past data. Therefore, to optimize the forgetting parameters $\boldsymbol{e}_t$, and $\boldsymbol{a}_t$ at time t, the proposed hypernetwork balances the current input data and the past latent value memory to store sufficient information of the learning history data before calculating the latent variables $\boldsymbol{M}_{t+1}^v$.

The proposed hypernetwork structure is located at the beginning of the Memory Updating Component (Figure 5). The inputs of the hypernetwork are the embedding vector $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$ and the past value memory $\tilde{\boldsymbol{M}}_t^v$. The embedding vector $\boldsymbol{v}_t$ is calculated from the current input data (s_j, u_{jt}) when a student responds to item j. In addition, $\tilde{\boldsymbol{M}}_t^v$ is calculated as

$$\tilde{\boldsymbol{M}}_t^v = \begin{cases} \boldsymbol{M}_t^v & (\lambda = 0), \\ \sigma(\boldsymbol{W}[\boldsymbol{M}_t^v, \boldsymbol{M}_{t-1}^v, \cdots, \boldsymbol{M}_{t-\lambda}^v] + \boldsymbol{\tau}) & (otherwise). \end{cases} \tag{42}$$

Where, $\boldsymbol{W}$ is the weight vector and $\boldsymbol{\tau}$ is the bias parameter vector. Next, $\boldsymbol{v}_t$ and $\tilde{\boldsymbol{M}}_t^v$ are optimized in the hypernetwork as

$$\tilde{\boldsymbol{v}}_t^{r'} = \delta_1 * \sigma(\boldsymbol{W}^v \tilde{\boldsymbol{M}}_t^{vr'-1}) \odot \boldsymbol{v}_t^{r'-1}, \tag{43}$$

$$\tilde{\boldsymbol{M}}_t^{vr'} = \delta_2 * \sigma(\boldsymbol{W}^M \tilde{\boldsymbol{v}}_t^{r'}) \odot \tilde{\boldsymbol{M}}_t^{vr'-1}, \tag{44}$$

where $\delta_1 \in \mathbb{R}$, $\delta_2 \in \mathbb{R}$, r is a hyperparameter and $1 \leq r' \leq r$. r represents the number of rounds in the recurrent architecture. If $r' = 1$, then $\tilde{\boldsymbol{v}}_t^0 = \boldsymbol{v}_t$ and $\tilde{\boldsymbol{M}}_t^{v0} = \tilde{\boldsymbol{M}}_t^v$. Because of the repeated multiplications in equations (43) and (44), this hypernetwork balances current data $\tilde{\boldsymbol{v}}_t$ and past value memory $\tilde{\boldsymbol{M}}_t^v$. For the proposed methods, we optimize the number of rounds r for each learning dataset. Details are presented in the Experiment section.

3.3.2 Memory Updating Component

Next, we estimate the forgetting parameters $\boldsymbol{e}_t$ and $\boldsymbol{a}_t$ using the optimized $\tilde{\boldsymbol{v}}^r$ and $\tilde{\boldsymbol{M}}_t^{vr}$. These forgetting parameters $\boldsymbol{e}_t$ and $\boldsymbol{a}_t$ are important to update the latest value memory $\boldsymbol{M}_{t+1}^v$ optimally. The earlier memory updating component of DKVMN and Deep-IRT calculates the forgetting parameters from $\boldsymbol{v}_t$ with only current input information in equations (37) and (38). By contrast, we calculate them using the optimized current input data $\tilde{\boldsymbol{v}}_t^r$ and the past latent value $\tilde{\boldsymbol{M}}_t^{vr}$. Furthermore, the unique feature of the proposed method is a new layer $\boldsymbol{z}_t$, which helps to optimize

$\boldsymbol{a}_t$. The memory updating component is located next to the hypernetwork on the upper right of Figure 5. The forgetting parameters $\boldsymbol{e}_t$ and $\boldsymbol{a}_t$ are calculated as

$$\boldsymbol{e}_t^{(l)} = \sigma(\boldsymbol{W}^{e1}\tilde{\boldsymbol{v}}_t^r + \boldsymbol{W}^{e2}\tilde{\boldsymbol{M}}_{t,l}^{vr} + \boldsymbol{\tau}^e), \tag{45}$$

$$\boldsymbol{z}_t^{(l)} = \sigma(\boldsymbol{W}^{z1}\tilde{\boldsymbol{v}}_t^r + \boldsymbol{W}^{z2}\tilde{\boldsymbol{M}}_{t,l}^{vr} + \boldsymbol{\tau}^z), \tag{46}$$

$$\boldsymbol{a}_t^{(l)} = \tanh(\boldsymbol{W}^{a1}\boldsymbol{z}_t^{(l)} + \boldsymbol{W}^{a2}\tilde{\boldsymbol{M}}_{t,l}^{vr} + \boldsymbol{\tau}^a). \tag{47}$$

Where, $\boldsymbol{W}^{(\cdot)}$ is the weight vector and $\boldsymbol{\tau}^{(\cdot)}$ is a bias vector. Then, the proposed method updates the latent value $\boldsymbol{M}_{t+1,l}^v$ as shown below.

$$\boldsymbol{M}_{t+1,l}^v = \tilde{\boldsymbol{M}}_{t,l}^{vr} \otimes (1 - w_{jl}\boldsymbol{e}_t^{(l)})^\top + w_{jl}\boldsymbol{a}_t^{(l)\top}. \tag{48}$$

By optimizing $\tilde{\boldsymbol{v}}_t$ and $\tilde{\boldsymbol{M}}_t^v$ in the hypernetwork, the parameters $\boldsymbol{e}_t$ and $\boldsymbol{a}_t$ are also estimated as optimizing the degree of forgetting of past data and as reflecting the current input data. Furthermore, the proposed method can capture the student knowledge state changes accurately because the latent knowledge state $\boldsymbol{M}_t^v$ has sufficient information related to the past learning history data.

4 Experiment of IRT Based on Deep Learning for Test Theory

4.1 Data Format

For test theory, we use a matrix of the students' responses as input data. The matrix size is $I \times J$ when the number of items is I, the number of students is J. Each element u_{ij} is 1 when the student i answers the item j correctly; it is 0 otherwise. All students address all items in the same order. If a student i does not address the item j, then the response u_{ij} is missing data.

4.2 Simulation Experiments

This section presents evaluation of the performances of the proposed method for test theory in Section 3.1 (designated as "Proposed") using simulation data according to earlier IRT studies of the linkage or the multi-population [87, 88]. We implemented Proposed using Chainer[1], a popular frameworks for neural networks. The values of tuning parameters are presented in Table 1. For implementation of IRT, we

[1]https://chainer.org/

employ 2PLM and estimate the parameters using EAP estimation with the MCMC algorithm.

For this experiment, we evaluate root mean square error (RMSE), Pearson's correlation coefficient, and the Kendall rank correlation coefficient between the estimated abilities and the true values. For calculation of RMSE, the estimated abilities of Proposed are standardized. The Kendall rank correlation coefficient is known to provide robust estimates for aberrant values.

Table 1: Values of tuning parameters.

Parameter	Value	Parameter	Value
$\lvert\boldsymbol{\theta}_1^{(i)}\rvert$	50	γ_1	0.1
$\lvert\boldsymbol{\theta}_2^{(i)}\rvert$	50	γ_2	0.1
$\lvert\boldsymbol{\beta}_1^{(j)}\rvert$	50	γ_3	0.1
$\lvert\boldsymbol{\beta}_2^{(j)}\rvert$	50	γ_4	0.1
Epoch	300	α_{L_e}	0.2
		α_{H_e}	0.8
		α_{L_i}	0.2
		α_{H_i}	0.8

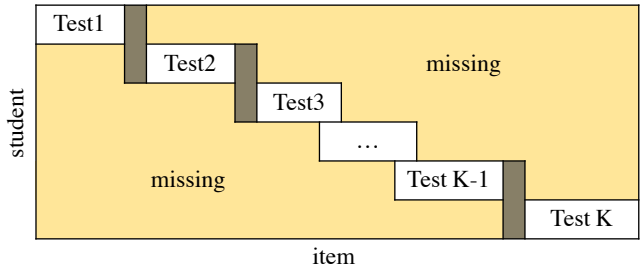


Figure 6: An example of item response pattern matrix.

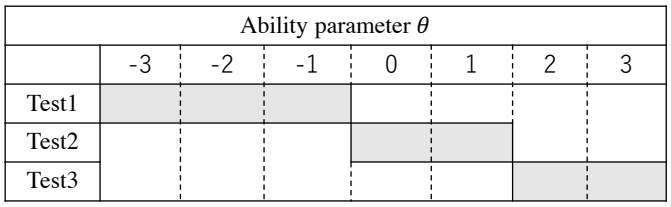


Figure 7: System generation.

4.2.1 Estimation Accuracy for Randomly Sampled Student Data

To underscore the effectiveness of Proposed for data of student' abilities that are not randomly sampled, this subsection presents evaluation of the estimation accuracy with changing student assignments for different tests. The procedures of this experiment are explained hereinafter.

This experiment generates 10 test data that have no common students. Also, the k-th test ($k = 1, ..., 10$) has common items only among the $k-1$-th test and the $k+1$-th test. Figure 6 shows an example of tests and the cells with brown hatching represent the common items. The true parameters were generated randomly:

$$\theta \sim \mathcal{N}(0, 1), \log a \sim \mathcal{N}(0, 1), b \sim \mathcal{N}(1, 0.4). \tag{49}$$

Here, the simulation data were generated based on 2PLM in the following two ways. The first way is that students are assigned randomly to each test according to their abilities generated from $\mathcal{N}(0, 1)$. The other way is that students are assigned systematically to each test as described below.

1. The students' abilities are sampled randomly from $\mathcal{N}(0, 1)$.
2. The students are sorted in order of their ascending ability. Furthermore, the students are divided equally into groups of 10 students in order of their respective abilities as shown in Figure 7.
3. The k-th student group is assigned to the k-th test.

Table 2 demonstrates the average of estimation accuracies under the condition of random generation. In Table 2, "No. items" represents the number of items for each test and "No. students" represents the number of students for each test. Results of the random assignment condition show that IRT outperforms Proposed. The reason is that the condition is an ideal situation for IRT because the data are generated randomly from the IRT model. However, for a small number of students or items, the differences between IRT and Proposed become smaller because the parameters can not be calculated accurately.

On the other hand, Table 3 demonstrates the average of estimation accuracies under the condition of system generation the results show that Proposed outperforms IRT for all datasets. Although, evaluating the abilities of numerous students on a single scale requires a linkage of students' abilities estimated from different tests, the linkage techniques of IRT assume random sampling of students' abilities from a standard normal distribution. Therefore, when this assumption does not guarantee,

IRT might not accurately estimate the students' abilities. In contrast, Proposed does not follow a standard normal distribution. As a result, Proposed suppresses the decline of accuracy in cases without common items among different tests. These results are expected to be beneficial for applying Proposed with actual data.

4.2.2 Estimation Accuracy for Multi-population Data

As described earlier, IRT assumes that students' abilities follow a standard normal distribution. Furthermore, it is known that no optimal linkage occurs under the assumption [89]. Additionally, no guarantee exists that students' abilities follow a standard normal distribution. When the assumption is violated, the ability estimation accuracy of IRT becomes extremely worse, even without the linkage problem. However, because Proposed does not assume random sampling from a statistical distribution, robust ability estimation is expected to be provided even when the IRT presumption is violated. To demonstrate the benefits of the proposed method, this subsection evaluates estimation accuracies of IRT and Proposed when students' abilities follow multiple populations.

For this experiment, the abilities of students taking different tests are assumed to be sampled from different populations. For this study, we assume two tests including 50 items. The abilities of the tests are sampled randomly from $N_1(\mu_1, \sigma^2)$ and $N_2(\mu_2, \sigma^2)$.For this experiment, the abilities of students taking different tests are assumed to be sampled from different populations. For this study, we assume two tests including 50 items. The abilities of the tests are sampled randomly from $\mathcal{N}_1(\mu_1, \sigma^2)$ and $\mathcal{N}_2(\mu_2, \sigma^2)$. Table 4 shows the average of estimation accuracies of students' abilities (RMSE) with different ability distributions and the number of common items. The standard deviation of each distribution was ascertained so that the total abilities' standard deviation is close to 1.0. Here, Wilcoxon's signed rank test is applied to infer whether the accuracies of IRT and Proposed are significantly different. The results showed that when the difference between μ_1 and μ_2 becomes small, IRT provides significantly high accuracy because the distribution approaches a single normal distribution. By contrast, as the difference between μ_1 and μ_2 becomes large, Proposed estimates student' abilities accurately. Therefore, Proposed is robust for estimation of student' abilities when they follow different distributions. The results also show that, when there is no common item, Proposed estimates the student' abilities more accurately than IRT does. Consequently, Proposed can estimate student' abilities accurately without common items.

Next, we demonstrate that Proposed can accommodate the abilities of multiple populations. Specifically, we generate abilities according to multiple populations for

Table 2: Parameter estimation accuracies. (randam)

Assignment	No. items	No. common items (Total no. items)	No. students (Total no. students)	Method	RMSE	Pearson	Kendall
random	10	5 (55)	50 (500)	Proposed	0.469	0.890	0.748
				IRT	**0.420**	**0.912**	**0.781**
			100 (1000)	Proposed	0.447	0.900	0.766
				IRT	**0.438**	**0.904**	**0.770**
			500 (5000)	Proposed	0.434	0.907	0.769
				IRT	**0.432**	0.907	**0.776**
			1000 (10000)	Proposed	0.424	0.908	**0.771**
				IRT	**0.411**	**0.911**	0.733
		0 (100)	50 (500)	Proposed	0.458	0.896	0.747
				IRT	**0.456**	0.896	**0.751**
			100 (1000)	Proposed	0.455	0.832	0.765
				IRT	**0.440**	**0.903**	**0.767**
			500 (5000)	Proposed	0.433	0.852	0.785
				IRT	**0.423**	**0.861**	**0.789**
			1000 (10000)	Proposed	0.412	0.910	**0.799**
				IRT	**0.403**	**0.914**	0.794
	30	5 (255)	50 (500)	Proposed	0.328	0.921	0.855
				IRT	**0.301**	**0.941**	**0.865**
			100 (1000)	Proposed	0.319	0.949	0.865
				IRT	**0.292**	**0.957**	**0.870**
			500 (5000)	Proposed	0.339	0.942	0.834
				IRT	**0.290**	**0.958**	**0.873**
			1000 (10000)	Proposed	0.329	0.947	0.844
				IRT	**0.298**	**0.968**	**0.879**
		0 (300)	50 (500)	Proposed	0.328	0.946	**0.860**
				IRT	**0.308**	**0.952**	0.858
			100 (1000)	Proposed	0.339	0.943	0.851
				IRT	**0.314**	**0.951**	**0.858**
			500 (5000)	Proposed	0.321	0.941	0.853
				IRT	**0.299**	**0.945**	**0.873**
			1000 (10000)	Proposed	0.302	0.938	0.853
				IRT	**0.281**	**0.948**	**0.881**
	50	5 (455)	50 (500)	Proposed	0.317	0.950	0.882
				IRT	**0.251**	**0.969**	**0.895**
			100 (1000)	Proposed	0.312	0.964	0.891
				IRT	**0.243**	**0.970**	**0.896**
			500 (5000)	Proposed	0.288	0.959	0.894
				IRT	**0.232**	**0.973**	**0.901**
			1000 (10000)	Proposed	0.278	0.961	0.894
				IRT	**0.234**	**0.973**	**0.901**
		0 (500)	50 (500)	Proposed	0.360	0.935	0.856
				IRT	**0.274**	**0.962**	**0.876**
			100 (1000)	Proposed	0.261	0.966	0.884
				IRT	**0.251**	**0.968**	**0.892**
			500 (5000)	Proposed	0.341	0.942	0.887
				IRT	**0.241**	**0.971**	**0.899**
			1000 (10000)	Proposed	0.266	0.968	0.889
				IRT	**0.241**	**0.972**	**0.901**

Table 3: Parameter estimation accuracies. (system)

Assignment	No. items	No. common items (Total no. items)	No. students (Total no. students)	Method	RMSE	Pearson	Kendall
system	10	5 (55)	50 (500)	Proposed	**0.665**	**0.778**	**0.568**
				IRT	1.111	0.381	0.237
			100 (1000)	Proposed	**0.622**	**0.807**	**0.629**
				IRT	0.779	0.696	0.466
			500 (5000)	Proposed	**0.611**	**0.812**	**0.639**
				IRT	0.792	0.702	0.499
			1000 (10000)	Proposed	**0.621**	**0.822**	**0.651**
				IRT	0.712	0.702	0.501
		0 (100)	50 (500)	Proposed	**0.997**	**0.502**	**0.267**
				IRT	1.170	0.314	0.184
			100 (1000)	Proposed	**0.721**	**0.740**	**0.561**
				IRT	1.176	0.308	0.197
			500 (5000)	Proposed	**0.701**	**0.761**	**0.591**
				IRT	1.016	0.498	0.277
			1000 (10000)	Proposed	**0.698**	**0.782**	**0.591**
				IRT	0.808	0.673	0.457
	30	5 (255)	50 (500)	Proposed	**0.561**	**0.835**	**0.696**
				IRT	0.613	0.786	0.622
			100 (1000)	Proposed	**0.501**	**0.875**	**0.716**
				IRT	0.573	0.836	0.672
			500 (5000)	Proposed	**0.499**	**0.878**	**0.722**
				IRT	0.553	0.846	0.679
			1000 (10000)	Proposed	**0.495**	**0.892**	**0.731**
				IRT	0.534	0.851	0.691
		0 (300)	50 (500)	Proposed	**0.661**	**0.781**	**0.586**
				IRT	0.786	0.691	0.489
			100 (1000)	Proposed	**0.579**	**0.832**	**0.664**
				IRT	0.762	0.709	0.506
			500 (5000)	Proposed	**0.561**	**0.852**	**0.684**
				IRT	0.732	0.705	0.512
			1000 (10000)	Proposed	**0.539**	**0.850**	**0.644**
				IRT	0.712	0.709	0.506
	50	5 (455)	50 (500)	Proposed	**0.376**	**0.929**	**0.802**
				IRT	0.426	0.909	0.760
			100 (1000)	Proposed	**0.393**	**0.923**	**0.811**
				IRT	0.805	0.750	0.543
			500 (5000)	Proposed	**0.372**	**0.930**	**0.810**
				IRT	1.044	0.454	0.282
			1000 (10000)	Proposed	**0.392**	**0.914**	**0.798**
				IRT	0.923	0.512	0.342
		0 (500)	50 (500)	Proposed	**0.635**	**0.798**	**0.599**
				IRT	0.782	0.694	0.489
			100 (1000)	Proposed	**0.408**	**0.916**	**0.785**
				IRT	0.612	0.812	0.532
			500 (5000)	Proposed	**0.421**	**0.891**	**0.765**
				IRT	0.598	0.822	0.495
			1000 (10000)	Proposed	**0.411**	**0.901**	**0.785**
				IRT	0.602	0.829	0.498

Table 4: Estimation accuracies (RMSE) for multi-population data.

No. students for each test	No. common items	μ_1	μ_2	σ^2	IRT	Proposed
500	5	-0.3	0.3	0.7	**0.186****	0.216
		-0.5	0.5	0.5	**0.184****	0.232
		-0.7	0.7	0.3	0.210	**0.206**
		-0.9	0.9	0.1	0.207	**0.195***
	0	-0.3	0.3	0.7	0.358	**0.325***
		-0.5	0.5	0.5	0.501	**0.324****
		-0.7	0.7	0.3	0.993	**0.382****
		-0.9	0.9	0.1	1.027	**0.385****
					**p<.01	*p<.05

data $N_1(-0.7, 0.3)$ and $N_2(0.7, 0.3)$ in Table 4. Figure 8 shows histograms of the true abilities, the estimated abilities using IRT, and the estimated abilities using Proposed. Figure 8 shows that Proposed clearly estimates a bimodal distribution as the ability distribution similar to the true distribution. The result demonstrates that Proposed flexibly expresses actual student' abilities distributions that do not follow a standard normal distribution.

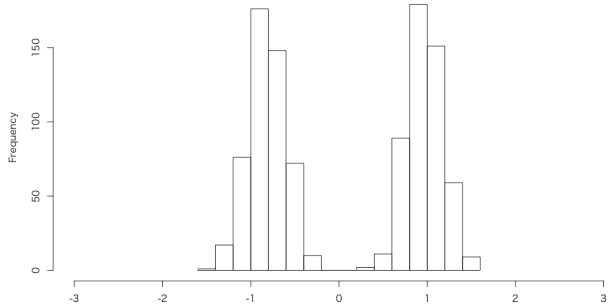
Next we evaluate the estimated ability distributions of IRT and Proposed using a fitting scores to the true distribution as

$$\sum_{k\in\{1,2\}} \sum_{i=1}^{I_k} \log p(\hat{\theta}_{ki}|\mu_k, \sigma), \tag{50}$$

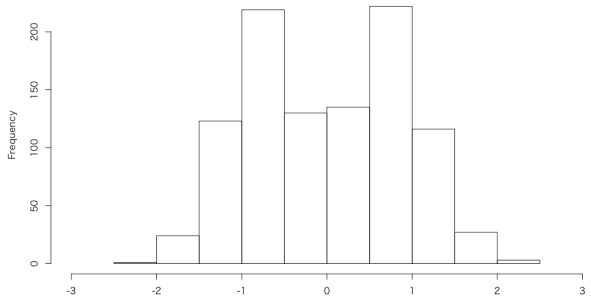
where I_k represents the number of students who took the k-th test. Also, $\hat{\theta}_{ki}$ is the estimated ability of i-th student for the k-th test. In addition, $p(\hat{\theta}_{ki}|\mu_k, \sigma)$ is the likelihood of estimated abilities given the true ability distribution as

$$p(\hat{\theta}_{ki}|\mu_k, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\hat{\theta}_{ki} - \mu_k)^2}{2\sigma^2}\right). \tag{51}$$

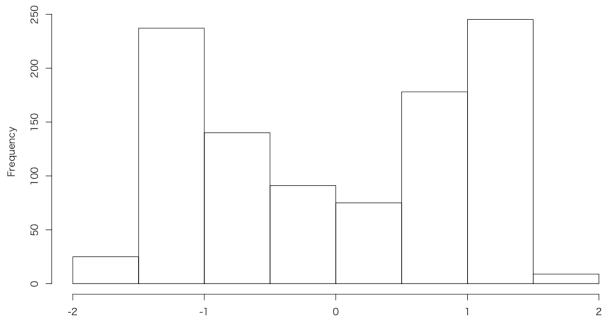
If the method fits the true distribution, then the estimated distribution approaches the true distribution. The fitting score of IRT is -1633.4. That of Proposed is -1437.1. The latter is higher than the former. Therefore, Proposed expresses the student' ability distributions more accurately than IRT does.



(a) True abilities



(b) Abilities estimated using IRT



(c) Abilities estimated using Proposed

Figure 8: Histograms of estimated abilities for multi-population data.

4.3 Actual Data Experiments

The simulation experiments suggested that Proposed might estimate student' abilities with high accuracy for actual data. This section evaluates the effectiveness of Proposed using the following actual datasets. Table 5 presents the number of students (No. Students), the number of items (No. Items), the average rate of items that a student did not address in the learning process (Rate.Sparse").

1. *Information* datasets consist of two test data (Information 1, 2) related to information technology. Information 1 has 169 students over 50 items. Information 2 has 266 students over 50 items. The tests were conducted of learning

management system, "Samurai" developed by [90, 91, 92]. Rate.Sparse is 0%.

2. *Critical thinking* dataset has 1221 undergraduate students over 179 items about critical thinking. Rate.Sparse is 87.8%.

3. *Program* datasets consist of two test data (Program 1, 2) about programming. Program 1 has 93 students over 13 items. Rate.Sparse is 0%. Program 2 has 74 students over 19 items with 6.8% Rate.Sparse.

4. *Practice Exam* dataset consists of two test data for high school students. Each test relates to mathematics and physics. Mathematics data have 12348 students over 48 items. Physics data have 9172 students over 24 items. The respective values of Rate.Sparse are 16.4% and 12.0%.

5. *CDM* datasets, which are widely used open datasets, are included in the R package CDM [93]. We used two datasets: ECPE and TIMSS. ECPE data include those for 2922 students over 28 language-related items. TIMSS data include those for 757 students over 23 math items. Rate.Sparse is 0%.

6. *Information Ethics* dataset has 31 undergraduate students over 90 items related to information ethics. Rate.Sparse is 46.3%.

7. *Engineer Ethics* dataset has 85 undergraduate students over 69 items related to engineer ethics. Rate.Sparse is 26.4%.

8. *Classi* datasets consist of three tests data for high school students: tests relate to physics, chemistry, and biology. The tests were conducted on the web-based system, "Classi[2]" using a tablet. Datasets have 239, 1139, and 192 students, respectively, and 119, 364, and 114 items. The respective values of Rate.Sparse are 92.4%, 96.4%, and 93.5%.

4.3.1 Reliability of Ability Estimation

This subsection presents evaluation of the reliability of abilities estimation of Proposed. Because the true values of parameters are unknown, we evaluate the reliabilities as follows. 1) Each dataset is divided equally into two sets of data. 2) Parameters of each method are estimated for the divided data from each dataset. 3) The RMSE and correlation between the two sets of the estimated parameters from the two divided datasets are calculated. 4) These procedures are repeated 10 times.

[2]https://classi.jp

Table 5: Summary of actual datasets.

Dataset	No. Students	No. Items	Rate.Sparse(%)
Information 1	169	50	0.0
Information 2	266	50	0.0
Critical Thinking	1221	179	87.8
Program 1	93	13	0.0
Program 2	74	19	6.8
Practice_Math	12348	48	16.4
Practice_Physics	9172	24	12.0
ECPE	2922	28	0.0
TIMSS	757	23	0.0
Information Ethics	31	90	46.3
Engineer Ethics	85	69	26.4
Classi_Physics	239	119	92.4
Classi_Chemistry	1139	364	96.4
Classi_Biology	192	114	93.5

The average of the RMSEs and correlations are calculated. Table 6 presents the results. Here, a Wilcoxon signed rank test is applied to infer whether the reliabilities of IRT and Proposed are significantly different.

Table 6 shows that Proposed provides more reliable ability estimates than IRT does. Especially, regarding the average of Kendall rank correlation coefficient, which is known to provide a robust estimate for aberrant values, Proposed outperforms IRT significantly. Results indicate that Proposed can estimate parameters more reliably than IRT does for actual test data. It is surprising that Proposed outperforms IRT for small datasets such as Program 1, Program 2, Statistics, Information Ethics, and Engineer Ethics. This result indicates Proposed as effective even for small datasets. For Practice_Math, and Practice_Physics, IRT has a higher Kendall rank correlation coefficient than Proposed does because the ability estimation of IRT tends to become stable when the dataset becomes large. IRT has that stability because it is guaranteed to converge asymptotically to the true joint probability distribution.

4.3.2 Prediction Accuracies for Student Performance

In the field of artificial intelligence in education, prediction of student's responses to unknown items from the student's past response history becomes important for adaptive learning systems [6, 8, 7, 59, 60]. This subsection presents comparison of the prediction accuracy of Proposed with that of IRT. Specifically, using ten-fold

Table 6: Reliability of ability parameter estimation.

Dataset	Method	RMSE	Pearson	Kendall
Information 1	IRT	**0.466**	**0.891**	0.685
	Proposed	0.514	0.867	**0.687**
Information 2	IRT	0.562	0.841	**0.668**
	Proposed	**0.555**	**0.845**	0.662
Critical Thinking	IRT	1.064	0.464	0.318
	Proposed	**1.025**	**0.474**	**0.327**
Program 1	IRT	0.890	0.599	0.403
	Proposed	**0.864**	**0.622**	**0.417**
Program 2	IRT	0.752	0.713	0.468
	Proposed	**0.720**	**0.737**	**0.475**
Practice_Math	IRT	**0.589**	**0.748**	**0.533**
	Proposed	0.744	0.723	0.514
Practice_Physics	IRT	**0.884**	**0.609**	**0.424**
	Proposed	0.911	0.585	0.411
ECPE	IRT	0.875	0.615	0.435
	Proposed	**0.874**	**0.618**	**0.440**
TIMSS	IRT	0.753	0.716	0.525
	Proposed	0.753	0.716	**0.523**
Information Ethics	IRT	0.394	0.920	0.643
	Proposed	**0.382**	**0.925**	**0.712**
Engineer Ethics	IRT	0.544	0.850	**0.403**
	Proposed	**0.517**	**0.865**	0.313
Classi_Physics	IRT	1.053	0.444	0.299
	Proposed	**0.943**	**0.554**	**0.403**
Classi_Chemistry	IRT	1.077	0.420	0.297
	Proposed	**0.923**	**0.574**	**0.439**
Classi_Biology	IRT	1.020	0.475	0.326
	Proposed	**0.748**	**0.717**	**0.531**
Average	IRT	0.764	0.680	0.451
	Proposed	**0.742**	**0.707**	**0.495***

* $p<0.05$

cross validation, the parameters are learned from training data and are used to predict responses in the remaining data. Then, we calculate the accuracy rates for the cross validation experiments. In this experiment, we use F1 score for the metric of the prediction accuracy. Here, a Wilcoxon signed rank test is applied to infer whether the respective accuracies of IRT and Proposed are significantly different.

Table 7 shows the results: the average of F1 value of Proposed is significantly higher than that of IRT. Proposed can predict student' responses to unknown items more accurately than IRT can. It is noteworthy that Proposed does not always outperform for large data. For Critical Thinking, IRT provides better performance than Proposed does, because Critical Thinking has high values of Rate.Sparse. Pro-

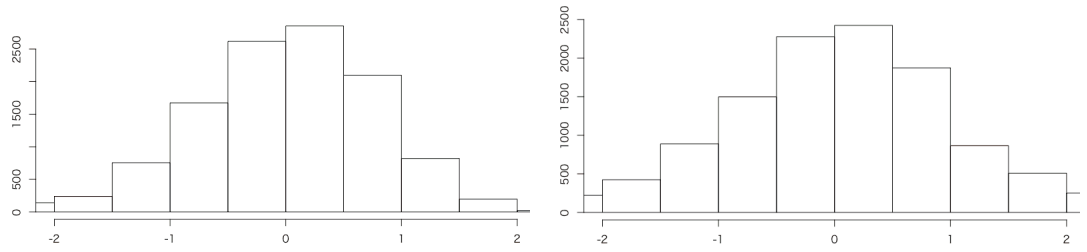
Table 7: Prediction accuracies of responses to unknown items.

Data	No. students	No. items	Rate.Sparse	IRT	Proposed
Information 1	169	50	0%	0.734	**0.737**
Information 2	266	50	0%	0.699	**0.700**
Critical Thinking	1221	179	87.8%	**0.695**	0.689
Program 1	94	13	0%	0.719	**0.729**
Program 2	74	19	6.8%	0.676	**0.685**
Practice_Math	12348	48	16.4%	**0.783**	0.780
Practice_Physics	9172	24	12.0%	**0.721**	0.710
ECPE	2922	28	0%	0.719	**0.729**
TIMSS	757	24	0%	0.711	**0.712**
Information Ethics	31	90	46.3%	0.746	**0.803**
Engineer Ethics	85	69	26.4%	0.634	**0.685**
Classi_Physics	239	119	92.4%	0.720	**0.721**
Classi_Chemistry	1139	364	96.4%	0.710	**0.711**
Classi_Biology	192	114	93.5%	0.722	**0.725**
Average				0.719	**0.728***

*p<.05

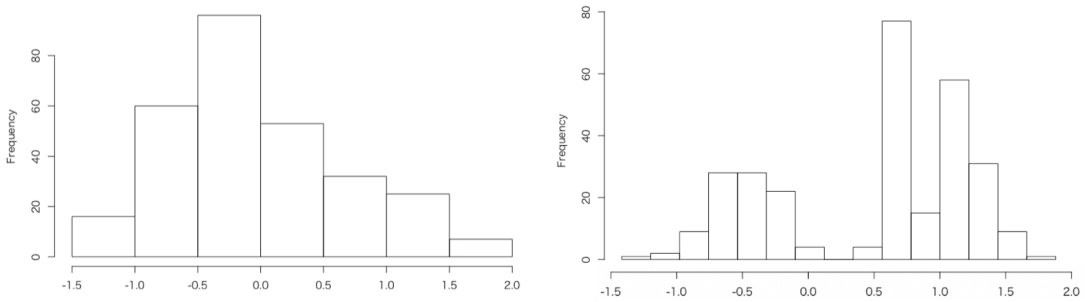
posed might be weak in dealing with sparse datasets. In contrast, for datasets with low values of Rate.Sparse, Proposed outperforms IRT even for small datasets. Generally speaking, the IRT prediction accuracy increases along with the number of students. Therefore, IRT has high prediction accuracies for Practice_Math and Practice_Physics.

Furthermore, Figure 9 depicts histograms of abilities estimated from Practice_Math, where the prediction accuracy of IRT is higher than that of Proposed. Figure 10 depicts histograms of abilities estimated from Classi_Biology data where the prediction accuracy of Proposed is higher than that of IRT. Figure 10 shows estimates conducted using both methods for the ability distribution similar to the standard normal distribution. In contrast, Figure 8 shows that Proposed expresses a multi modal distribution, although IRT estimates a unimodal distribution. Proposed can predict responses to unknown items because it can flexibly express distributions of various abilities.



(a) Abilities estimated using IRT (b) Abilities estimated using Proposed

Figure 9: Histograms of abilities estimated using IRT and Proposed for Practice_Math data.



(a) Abilities estimated using IRT (b) Abilities estimated using Proposed

Figure 10: Histograms of abilities estimated using IRT and Proposed for Classi_Biology data.

5 Experiment of IRT Based on Deep Learning for Knowledge Tracing

5.1 Data Format

For KT, we use time-series data consisting of the students' responses collected from the online learning systems. The students' response u_{ij} is 1 when the student i answers the item j correctly; it is 0 otherwise. The students address similar items repeatedly to master a certain concept. Therefore, the numbers of responses differs among students. Furthermore, we use item and skill tags for input data to estimate the relation between each item and skill.

5.2 Prediction Accuracies for Student Performance

In the preceding section, we showed that the proposed method for test theory has higher parameter interpretability and prediction accuracy for students' performance than the standard IRT model has. As described in this section, we conduct experiments to compare the performances of the proposed method for KT in Section 3.2 (designated as "Proposed-KT") and the proposed method with a hypernetwork in Section 3.3 (designated as "Proposed-HN") against existing solutions. This section presents a comparison of the prediction accuracies for student performance of the proposed methods with those of earlier methods (DKVMN [8], Deep-IRT [7], AKT [12]) using six benchmark datasets as ASSISTments2009[3], ASSISTments2015[4], ASSISTments2017[5], Statics2011[6], Junyi[7], Eedi[8]. The details of the dataset are as follows.

1. *ASSISTments* datasets (ASSISTments2009, ASSISTments2015, and ASSISTments2017) collected from online learning systems have been used as the standard benchmark for KT methods.

2. *Statics2011* dataset was collected from college-level engineering courses on statics.

3. *Junyi* dataset was collected by Junyi Academy, a Chinese online learning system [94]. We use only the students' exercise records in the math curriculum. Additionally, we select items that the students attempted for the first time without hints. We also changed the question types into unique skill number tags.

4. *Eedi* dataset includes data from the school years of 2018–2020, with student responses to mathematics questions from Eedi, a leading educational platform by which millions of students interact daily around the globe [95]. For Eedi, each item has a list of hierarchical knowledge components. We convert these lists into unique skill number tags.

ASSISTments2009, ASSISTments2017, and Eedi have item and skill tags, although most methods explained in the relevant literature adopt only the skill tag

[3]https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data
[4]https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data
[5]https://sites.google.com/view/assistmentsdatamining
[6]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
[7]http://www.junyiacademy.org/
[8]https://eedi.com/projects/neurips-education-challenge

as an input. However, methods with skill inputs rely on the assumption that items with the same skill are equivalent [12]. That assumption does not hold when an item's difficulties in the same skill differ greatly. Therefore, as inputs to AKT and the proposed method, we employ not only skills but also items [12, 13, 45]. Also, for ASSISTments2015, Statics2011, and Junyi with only skill tags, we employ the skill as input data. Table 8 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).

These datasets include the numerous students' responses to the numerous items in the long learning process. Although the online learning system helps students to learn effectively by presenting the optimal item for adaptive learning, it is difficult to choose the optimal item only from the student ability parameters and the item parameters. Therefore, predicting a student's response is important to identify the optimal problem and to discover concepts that the student has not mastered.

Table 8: Summary of benchmark datasets.

Dataset	No. students	No. skills	No. Items	Rate Correct	Learning length
ASSISTments2009	4151	111	26684	63.6%	52.1
ASSISTments2015	19840	100	N/A	73.2%	34.2
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

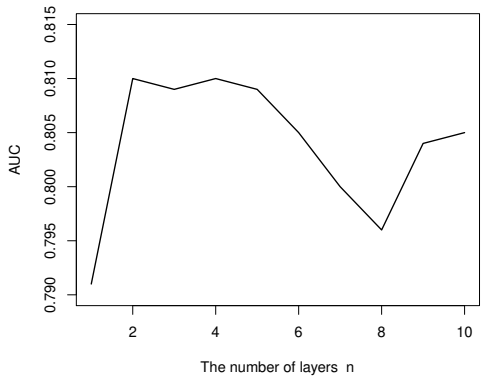


Figure 11: AUC and the number of layers for ASSISTments2009. The vertical axis shows AUC on the left side. The horizontal axis shows the number of layers.

5.3 Hyperparameter Selection and Evaluation

We used standard five-fold cross-validation to evaluate the respective prediction accuracies of the methods. According to Ghosh et al. (2020) [12], for each fold, 20% learners are used as the test set, 20% are used as the validation set, and 60% are used as the training set. We employ Adam optimization with a learning rate of 0.003 and batch-size 32. For all methods, the hidden layer size and memory dimension are chosen from $\{10, 20, 50, 100, 200\}$ using cross-validation. In addition, for the earlier methods, we used the hyperparameters reported from earlier studies [7, 12]. Additionally, we set 200 items as the upper limit of the input length according to an earlier study [7, 8]. When the input length of items is greater than 200, we use the first 200 response data for all methods.

To ascertain the number of layers k for the proposed method, we conducted some experiments to gain experience using ASSISTments2009 while changing the layer number. The results are presented in Figure 11. As the figure shows, the AUC score reaches its highest value when $k = 2$ and $k = 4$. Based on this finding, we employ $k = 2$ for the following experiments because the computation time of the proposal increases exponentially as the number of layers increases.

If the calculated 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 three metrics for prediction accuracy: Accuracy (Acc) score, AUC score, and Loss score. The first, Acc represents the concordance rate between the student predictive responses and the actual responses. The second, AUC provides a robust metric for binary prediction evaluation. When an AUC score is 0.5, the prediction performance is equal to that of random guessing. Loss represents the cross-entropy in equation (28).

5.4 Hyperparameter Selection in Hypernetwork

5.4.1 Optimal Tuning Parameter δ_1 and δ_2 Estimation

For our experiments, we optimize the δ_1 and δ_2 to adjust the hypernetwork for each dataset. To choose the optimal parameters δ_1 and δ_2, we conducted some experiments using all training datasets by changing δ_1 and δ_2, respectively. The optimal tuning parameters $\{\delta_1, \delta_2\}$ are estimated as $\{1.5, 1.5\}$ for ASSISTments2009, ASSISTments2015 and ASSISTments2017, $\{1.0, 1.7\}$ for Statics2011, $\{1.0, 1.0\}$ for Junyi and Eedi. Based on this result, we employ these tuning parameters for the

Table 9: Prediction accuracy and hyperparameter r.

Dataset	Number of rounds r					
	2	3	4	5	6	7
Statics2011 (skill)	**82.25**	82.24	82.20	82.20	82.16	82.11
ASSISTments2009 (skill)	81.19	**81.83**	81.25	81.23	81.2	80.96
ASSISTments2015 (skill)	72.91	**72.95**	72.90	72.89	72.81	72.73
ASSISTments2017 (skill)	**85.06**	82.73	81.64	80.17	73.23	72.64
Junyi (skill)	**79.00**	78.74	78.71	78.67	78.62	78.65
Eedi (skill)	**75.53**	N/A	N/A	N/A	N/A	N/A
ASSISTments2009 (item & skill)	81.30	81.14	81.38	81.49	**82.55**	81.20
ASSISTments2017 (item & skill)	75.94	76.17	76.74	76.70	**76.85**	76.74
Eedi (item & skill)	**79.27**	N/A	N/A	N/A	N/A	N/A

following experiments.

5.4.2 Optimal Number of Rounds r Estimation

To ascertain the number of rounds r in the hypernetwork, we conducted some experiments to gain experience using the training datasets by changing the value of r. The results are presented in Table 9. As the table shows, the numbers of rounds r are estimated as $r = 2$ for Statics2011, ASSISTments2017 and Junyi with skill inputs, as $r = 3$ for ASSISTments2009 and ASSISTments2015 with skill inputs and as $r = 6$ for ASSISTments2009 and ASSISTments2017 with item and skill inputs. For Eedi dataset with numerous students, the proposed method can not complete the calculation because of exploding gradients under the condition of $r = 3$ or more.

The prediction accuracy of the proposed method tends to follow a convex function of the number of rounds r. Therefore, the proposed method estimates the number of each round r by incrementing the value from the initial value $r = 2$ to maximize the prediction accuracy. Especially, the optimal r estimates for ASSISTments2009

and ASSISTments2017 with item and skill inputs provide large values. The optimal value of r might be related to the number of input data.

5.4.3 Optimal Degree of Past Latent Variables to be Assessed

The input of the hypernetwork $\tilde{\boldsymbol{M}}_t^v$ is calculated from the past latent variables $\{\boldsymbol{M}_t^v, \boldsymbol{M}_{t-1}^v, \cdots, \boldsymbol{M}_{t-\lambda}^v\}$ at time $t-\lambda$ to t. We optimize λ by changing the value of $\lambda \in \{0, 1, 2, \cdots, t\}$ using the optimal δ_1, δ_2, and r for each learning dataset. Results show that the optimal λ can be estimated as $\lambda = 1$ for ASSISTments2009 and Junyi with skill inputs, and as ASSISTments2009 and ASSISTments2017 with item and skill inputs. When using the other datasets, optimal λ is estimated as $\lambda = 0$.

5.5 Results

5.5.1 Skill Inputs

The respective values of Acc, AUC, and Loss for all benchmark datasets with only skill inputs are presented in Table 10. Additionally, this report describes the standard deviations across five test folds. Proposed-KT and Proposed-HN respectively represent variants of the proposed method with and without the hypernetwork.

Results show that the averages of AUC, ACC, and Loss obtained using Proposed-KT are better than those using Deep-IRT, although the proposed method separates student and item networks. This result implies that redundant deep student and item networks function effectively for performance prediction. These results are explainable from reports of state-of-the-art methods [42, 43, 44].

Also, Proposed-HN, which optimizes the forgetting parameters in the hypernetwork, provides the best average scores for all metrics. Proposed-HN improves the prediction accuracy of Proposed-KT. However, the performances of Proposed-HN and AKT were found to have no significant difference in multiple comparison tests. The findings suggest that the Proposed-HN performs comparably to AKT, which reportedly has the highest accuracy among the earlier methods. For each dataset, results indicate that Proposed-HN provides the best AUC scores for ASSISTments2009, ASSISTments2017, Statics2011, and Junyi. Especially for ASSISTments2017 with long learning lengths, the performance of the Proposed-HN markedly outperforms that of AKT. By contrast, Proposed-HN tends to have lower prediction accuracies for ASSISTments2015 with a shorter learning length than AKT has. Results suggest that the proposed hypernetwork functions effectively, especially for datasets with long learning lengths.

Table 10: Prediction accuracies of student's performance with skill inputs.

Dataset	metrics	DKVMN	Deep-IRT	AKT	Proposed-KT	Proposed-HN
ASSISTments2009	AUC	81.21 +/- 0.31	81.34 +/- 0.39	80.81 +/- 0.41	81.34 +/- 0.24	**81.83 +/- 0.30**
	Acc	75.11 +/- 0.66	76.55 +/- 0.45	76.57 +/- 0.55	**76.91 +/- 0.24**	76.80 +/- 0.49
	Loss	0.47 +/- 0.05	0.48 +/- 0.10	0.49 +/- 0.08	0.47 +/- 0.10	**0.46 +/- 0.11**
ASSISTments2015	AUC	72.61 +/- 0.16	72.53 +/- 0.23	**72.97 +/- 0.12**	72.34 +/- 0.13	72.95 +/- 0.14
	Acc	75.05 +/- 0.18	74.97 +/- 0.14	**75.25 +/- 0.10**	74.95 +/- 0.39	75.02 +/- 0.15
	Loss	0.51 +/- 0.02	0.52 +/- 0.03	**0.51 +/- 0.01**	0.52 +/- 0.02	**0.51 +/- 0.03**
ASSISTments2017	AUC	72.67+/- 0.37	72.08 +/- 0.32	73.25+/- 0.41	72.32+/- 0.69	**85.06 +/- 1.17**
	Acc	68.46 +/- 0.24	68.36 +/- 0.30	69.17+/- 0.70	68.07 +/- 0.54	**79.11 +/- 1.06**
	Loss	0.58+/- 0.03	0.59 +/- 0.07	0.58+/- 0.09	0.60 +/- 0.08	**0.48 +/- 0.24**
Statics2011	AUC	81.20 +/- 0.42	81.38 +/- 0.42	82.15 +/- 0.35	81.45 +/- 0.45	**82.25 +/- 0.55**
	Acc	79.24 +/- 0.84	80.33 +/- 0.78	80.41 +/- 0.67	79.18 +/- 0.67	**80.63 +/- 0.85**
	Loss	0.42 +/- 0.14	0.42 +/- 0.18	0.42 +/- 0.13	0.42 +/- 0.12	**0.41 +/- 0.20**
Junyi	AUC	78.59 +/- 0.21	78.39 +/- 0.20	78.84 +/- 0.19	78.47 +/- 0.21	**79.00 +/- 0.26**
	Acc	86.61 +/- 0.28	86.57 +/- 0.30	86.54 +/- 0.25	86.58 +/- 0.27	**86.76 +/- 0.24**
	Loss	0.31 +/- 0.07	0.31 +/- 0.07	0.31 +/- 0.04	0.31 +/- 0.06	**0.30+/- 0.05**
Eedi	AUC	75.11 +/- 0.16	75.63 +/- 0.17	**75.81 +/- 0.15**	75.76 +/- 0.17	75.53 +/- 0.15
	Acc	71.23 +/- 0.24	71.34 +/- 0.29	71.38 +/- 0.20	**71.41 +/- 0.25**	71.30 +/- 0.24
	Loss	0.59 +/- 0.06	**0.56 +/- 0.07**	**0.56 +/- 0.03**	**0.56 +/- 0.06**	0.57 +/- 0.06
Average	AUC	76.89	76.83	77.30	76.91	**79.35**
	Acc	74.46	75.05	76.55	76.18	**78.27**
	Loss	0.48	0.48	0.48	0.48	**0.46**

To investigate the reason for that phenomenon, we analyze the forgetting parameters $\boldsymbol{e_t}$ and $\boldsymbol{a_t}$ in the memory updating component of the proposed method. As described above, $\boldsymbol{e}_t$ influences the degree to which the value memory forgets the past ability. Also, $\boldsymbol{a}_t$ controls how much the value memory reflects the current input data. We calculate the l_2-norm of the forgetting parameters $\boldsymbol{e_t}$ and $\boldsymbol{a_t}$ for the earlier memory updating component (of Proposed-KT) and the new memory updating component with hypernetwork (of Proposed-HN), respectively using the ASSISTments2017 dataset. This experiment is not aimed at comparing the predictive accuracies but at analyzing the parameter estimators within the memory updating component. Furthermore, each method is tuned to maximize the performance prediction accuracy (AUC) of the validation set. Table 11 presents the averages of the l_2-norms of $\boldsymbol{e_t}$ and $\boldsymbol{a_t}$ at time $t \in \{1, 2, \cdots, T\}$. Table 11 shows that Proposed-KT has the larger l_2-norm value of $\boldsymbol{e_t}$ than $\boldsymbol{a_t}$. The earlier memory updating component drastically forgets the student's past ability information and reflects the current input data when the latent variable memory is updated. The reason is that the forgetting parameters $\boldsymbol{e_t}$ and $\boldsymbol{a_t}$ are calculated using only the current input data. Therefore, their latent value memory $\boldsymbol{M}_t^v$ might not store the

Table 11: Forgetting parameters' norm average.

norm average	Proposed-KT	Proposed-HN
$\|\boldsymbol{e_t}\|$	5.12	1.99
$\|\boldsymbol{a_t}\|$	3.17	2.58

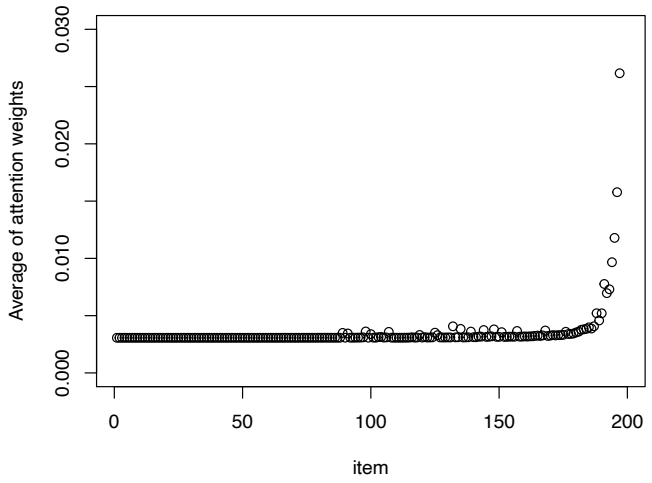


Figure 12: Average of attention weights in AKT for ASSISTments2017.

student's past ability information. By contrast, Proposed-HN has a larger l_2-norm value of $\boldsymbol{a_t}$ than $\boldsymbol{e_t}$. In the memory updating component of Proposed-HN, $\boldsymbol{a_t}$ and $\boldsymbol{e_t}$ are calculated using both the current input data $\boldsymbol{v_t}$ and the past latent value memory $\boldsymbol{M}_t^v$. Furthermore, these $\boldsymbol{v_t}$ and $\boldsymbol{M}_t^v$ are optimized in the hypernetwork to balance both the current input data and the student's past ability information. The results obtained for the other datasets are almost identical to those obtained for ASSISTments2017, although they are omitted to avoid redundant descriptions. Therefore, results suggest that the Proposed-HN works more effectively for long learning processes because hypernetwork facilitates the reflection of past data.

Findings indicate that AKT provides the best performance for ASSISTments2015. However, the AKT performance results are worse than those of Proposed-HN for ASSISTments2017. Figure 12 shows the average of attention weights of all students for the 200 items in ASSISTments2017. The vertical axis presents the average of attention weights. The horizontal axis presents the number of items the student addressed. Figure 12 shows that the attention weight α decays as the distance between the current input time and the past input time increases. It is noteworthy

that the attention weight α converges to a certain non-zero value. This finding implies that AKT does not completely forget even past data obtained at an extremely long time prior. Consequently, AKT might inadequately forget the past response data from long learning processes. However, Gosh et al. (2020) reported that AKT is more effective for large datasets. Therefore, AKT provides the best performance for AUC of Eedi, which has an extremely large number of students. The performance results obtained using DKVMN are almost identical to those obtained using Deep-IRT because they have similar network structures.

5.5.2 Item and Skill Inputs

Furthermore, we compared the performances of the proposed methods with those of AKT for ASSISTments2009, ASSISTments2017, and Eedi with item and skill inputs according to [12]. The respective values of Acc, AUC, and Loss are presented in Table 12. Results indicate that the Proposed-HN provides the best performance for the all metrics: averages of AUC, Acc, and Loss. For each dataset, the Proposed-HN provides the best scores for ASSISTments2009 and for ASSISTments2017. As described above, the Proposed-HN greatly outperforms AKT for ASSISTments2017 with a long learning length because the proposed hypernetwork functions effectively. However, for Eedi, AKT provides the best scores for all the metrics. In fact, AKT with item and skill inputs provides higher performance than those achieved using only skill inputs, as shown in [12]. In contrast, the proposed methods with item and skill inputs do not necessarily outperform those with only skill inputs. The reason might be that input item information cannot be used effectively because the latent value memory $\boldsymbol{M}_t^v$ is optimized using only input skills in the memory updating component. In addition, for Eedi, because of the increased number of parameters, it might not completely tune the hyperparameters in the hypernetwork.

Moreover, we experimented with Temporal IRT (TIRT) [20, 21]. It is a Hidden Markov IRT with a parameter to forget past response data, as described earlier in section 2.1. IRT-based methods rely on an assumption of local independence among the student item responses. They should not be applied to learning processes that allow a student to respond to the same item repeatedly. Therefore, we employ not skills but items as inputs using ASSISTments2009 and ASSISTments2017. Additionally, we respectively decompose these datasets into each skill group and estimate the parameters from skill data independently because TIRT assumes a single dimension skill of the ability. In other words, TIRT predicts performance using only an ability corresponding to one skill for an item. To estimate the student ability and item parameters of TIRT, we employ the expected a posteriori (EAP) estimators

Table 12: Prediction accuracies of student performance with item and skill inputs.

Dataset	metrics	AKT	Proposed-KT	Proposed-HN
ASSISTments2009	AUC	82.20 +/- 0.25	80.70 +/- 0.56	**82.55 +/- 0.32**
	Acc	77.30 +/- 0.55	76.13 +/- 0.58	**77.42 +/- 0.49**
	Loss	0.49 +/- 0.10	0.54 +/- 0.10	**0.47 +/- 0.11**
ASSISTments2017	AUC	74.54+/- 0.21	74.15+/- 0.27	**77.69 +/- 0.51**
	Acc	69.83+/- 0.15	68.73+/- 0.11	**72.16 +/- 0.55**
	Loss	0.58+/- 0.06	0.57+/- 0.06	**0.54 +/- 0.13**
Eedi	AUC	**79.42 +/- 0.11**	79.11 +/- 0.14	79.27 +/- 0.15
	Acc	**73.59 +/- 0.16**	73.42 +/- 0.24	73.49 +/- 0.27
	Loss	**0.52 +/- 0.02**	0.53 +/- 0.00	0.53 +/- 0.00
Average	AUC	78.72	78.00	**79.83**
	Acc	73.57	72.76	**74.36**
	Loss	0.53	0.55	**0.51**

using the Markov chain Monte Carlo (MCMC) method. The results indicate that AUC is 80.38, Acc is 76.39, and Loss is 0.49 for ASSISTments2009. For ASSISTments2017, results show that AUC is 75.52, Acc is 84.71, and Loss is 0.46. Surprisingly, TIRT outperforms AKT with skill input for ASSISTments2017. That finding suggests that TIRT might estimate the student ability transition accurately. For the Eedi dataset, TIRT can not complete the calculations within 24 hour because of its data size.

6 Parameter Interpretability

6.1 Estimation Accuracy of Ability Parameters

In the preceding section, we showed that the proposed method has higher prediction accuracy than other methods have. As described in this section, to evaluate the interpretability of the ability parameters of the proposed method, we use simulation data to compare the parameter estimates with those of Deep-IRT [7]. These datasets are generated from TIRT [20, 21]. 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. Therefore, as ϵ increases, the fluctuation range of the true ability increases at each time point. For this experiment, the priors of the j-th item parameters are $\log a_j \sim \mathcal{N}(0, 1)$, $b_j \sim \mathcal{N}(0, 1)$. Each

Table 13: 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	Deep-IRT	0.626	0.667	0.740	0.738	0.626	0.660	0.750	0.745	0.441	0.473	0.550	0.549
	Proposed-KT	0.885	0.907	0.924	0.916	0.892	0.915	0.940	0.938	0.710	0.746	0.785	0.782
	Proposed-HN	**0.902**	**0.916**	**0.930**	**0.927**	**0.910**	**0.923**	**0.943**	**0.941**	**0.736**	**0.761**	**0.790**	**0.792**
0.3	Deep-IRT	0.730	0.799	0.808	0.823	0.751	0.831	0.862	0.873	0.551	0.628	0.659	0.670
	Proposed-KT	0.827	0.891	0.883	0.890	0.863	0.926	0.941	0.945	0.671	0.755	0.778	0.785
	Proposed-HN	**0.840**	**0.905**	**0.900**	**0.907**	**0.877**	**0.932**	**0.947**	**0.954**	**0.689**	**0.767**	**0.791**	**0.804**
0.5	Deep-IRT	0.773	0.800	0.807	0.814	0.812	0.861	0.877	0.890	0.605	0.654	0.676	0.692
	Proposed-KT	0.855	0.870	0.860	0.849	0.893	0.928	0.929	0.930	0.705	0.755	0.758	0.761
	Proposed-HN	**0.874**	**0.871**	**0.869**	**0.859**	**0.901**	**0.928**	**0.934**	**0.940**	**0.720**	**0.755**	**0.768**	**0.779**
1.0	Deep-IRT	0.788	0.809	0.824	0.813	0.834	0.884	0.891	0.888	0.626	0.684	0.695	0.692
	Proposed-KT	0.843	0.830	0.844	0.834	0.886	0.911	0.919	0.918	0.696	0.728	0.740	0.740
	Proposed-HN	**0.854**	**0.840**	**0.854**	**0.836**	**0.894**	**0.920**	**0.930**	**0.919**	**0.708**	**0.744**	**0.762**	**0.743**

dataset includes 2000 student responses to $\{50, 100, 200, 300\}$ items. The discrimination parameter $\boldsymbol{a}$ and the item's difficulty parameter $\boldsymbol{b}$ are estimated using 1800 students' response data. Given the estimated $\boldsymbol{a}$ and $\boldsymbol{b}$, we estimate the students' ability parameters using the remaining 200 students' response data. In addition, for each dataset, we obtain results while changing $\epsilon = \{0.1, 0.3, 0.5, 1.0\}$.

We evaluate the Pearson's correlation coefficients, the Spearman's rank correlation coefficients, and the Kendall rank correlation coefficients between the true ability parameters of the true model (TIRT) and the estimated ability parameters of the KT methods (Deep-IRT, Proposed-KT, and Proposed-HN). The Spearman's rank correlation is the nonparametric version of Pearson's correlation. The Kendall rank correlation coefficient is known to provide robust estimates for aberrant values [46]. Generally, the estimation accuracy of the ability parameters is evaluated using root mean square error (RMSE). However, a student's ability of TIRT does not assume a standard normal distribution because the student ability distribution differs at each time. We are unable to evaluate RMSE in this experiment because TIRT, Deep-IRT [7], and the proposed methods are unable to not standardize their student abilities.

We calculate a correlation coefficient using a student's abilities θ_t at time $t \in \{1, 2, \cdots, T\}$, as estimated using TIRT and the KT methods (Deep-IRT, Proposed-KT, and Proposed-HN). Next, we average these correlation coefficients of all students. Table 13 presents the average correlation coefficients of the methods for the respective conditions. Results show that, for all conditions, Proposed-KT and Proposed-HN provide stronger correlation with the true ability parameters than

Deep-IRT does. The results of Spearman's rank correlation coefficients of the proposed method are greater than those of Pearson's correlation coefficients because the student's ability distribution changes constantly over time in TIRT. Especially, the results obtained for the Kendall rank correlation coefficients suggest that Proposed-KT and Proposed-HN estimate the abilities robustly, even for aberrant values. The results demonstrate that the two independent networks proposed function effectively to provide appropriate interpretability of the estimated parameters. Moreover, the students' ability parameters are estimated accurately with sufficient information from past learning history data because the hypernetwork optimized the forgetting parameters using both current input data and past data. Furthermore, the proposed methods tend to produce stronger correlations as the number of items increases. These findings suggest that the proposed methods represent the true student's ability transition accurately in long learning processes.

6.2 Ability Estimate Characteristics Analyses

This section we analyze the ability estimates of Proposed-KT and compare them with those of Proposed-HN. [13]. To compare the parameter characteristics, we calculate the following two metrics.

1. Intra-individual variance: V_i denotes the variance of student i's abilities during student i's learning process ($t \in \{1, \cdots, T_i\}$). V stands for the average of all students' V_i.

$$\bar{\theta}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \theta_i^t, \tag{52}$$

$$V_i = \frac{1}{T_i} \sum_{t=1}^{T_i} (\theta_i^t - \bar{\theta}_i)^2, \tag{53}$$

$$V = \frac{1}{I} \sum_{i=1}^{I} V_i. \tag{54}$$

   Therein, θ_i^t represents the time point of the ability $\theta^{(t,j)}$ of student i at time t.

2. Inter-individual variance: $V_t^{'}$ is the variance of all students' abilities at time t.

$V^{'}$ is the average of all students' $V_t^{'}$.

$$\bar{\theta^t} = \frac{1}{I}\sum_{i=1}^{I}\theta_i^t, \tag{55}$$

$$V_t^{'} = \frac{1}{I}\sum_{i=1}^{I}(\theta_i^t - \bar{\theta^t})^2, \tag{56}$$

$$V^{'} = \frac{1}{T_i}\sum_{t=1}^{T_i}V_t^{'}. \tag{57}$$

We use ASSISTments2017 with only skill inputs and item and skill inputs to demonstrate that Proposed-HN works effectively in the long learning process. ASSISTments2017 has the longest learning length in the datasets (shown in Table 10 and Table 12). Table 14 shows the Inter-individual variances V of the students' ability estimated by Proposed-KT and Proposed-HN. As a result, Proposed-HN has larger inter-individual variances than Proposed-KT has. The larger inter-individual variance means that the model discriminates for each student's ability well. Therefore, Proposed-HN can distinguish more accurately the students' abilities than Proposed-KT can.

Next, Table 15 shows the intra-individual variances $V^{'}$ of the students' ability estimated by Proposed-HN and Proposed-KT. Proposed-HN has larger intra-individual variances than Proposed-KT has. The large intra-individual variance signifies that the range of a student's ability transition is wide. That is to say, the ability estimates of Proposed-HN fluctuate largely over a wide range. On the other hand, Proposed-KT has small intra-individual variances. When the students address the items in the long learning process, the ability estimate of Proposed-KT might converge to a certain value because its memory updating component is not optimized.

These results demonstrated that Proposed-HN can accurately capture each student's skill abilities change. This advantage is important to construct a student model for adaptive learning,

Table 14: Inter-individual variances.

Inputs	Proposed-KT	Proposed-HN
skill	0.1020	2.2128
item&skill	0.0503	0.0896

Table 15: Intra-individual variances.

Inputs	Proposed-KT	Proposed-HN
skill	0.0704	1.9631
item&skill	0.0339	0.0724

6.3 Student Ability Transitions

This section shows student ability transitions using the proposed method (Propsoed-HN). Visualizing the ability transition for each skill is helpful for both students and teachers because they can reveal student strengths and weaknesses and can improve the learning method to fill in the learning gaps. Yeung (2019) [7] demonstrated a student ability transition for each skill using Deep-IRT. However, their results included some counter-intuitive ability estimates. For example, even when the student answered incorrectly, the corresponding student ability estimate increased. Moreover, Deep-IRT cannot identify a relation among multidimensional skills. In some cases, a student's ability for low-level skills decreases even when the student responds correctly to items for high-level skills.

Fig. 13 depicts an example of student ability transitions of each skill estimated using Deep-IRT and Proposed-HN for the ASSISTments2009 according to earlier studies [7, 45]. The vertical axis shows the student's ability value on the right side. 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. In the first 30 attempts, the student attempted skills of "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).

For Deep-IRT, as in earlier reports [7], some part of ability changes might be inconsistent with response data. For instance, the ability of skill "equation solving more than two steps" (grey), which is a higher-level skill, decreases even though the student responds correctly to items 11–17. In another instance, the student responds correctly to items for high-level skills even when a student's ability for low-level skills "equation solving two or few steps" (green) decreases. These unstable behaviors of Deep-IRT might engender severe difficulties, which will consequently confuse students and teachers, as a student model.

In contrast, Fig. 13 indicates that Proposed-HN can provide accurate estimates to reflect the student responses. Additionally, it can estimate relations among the

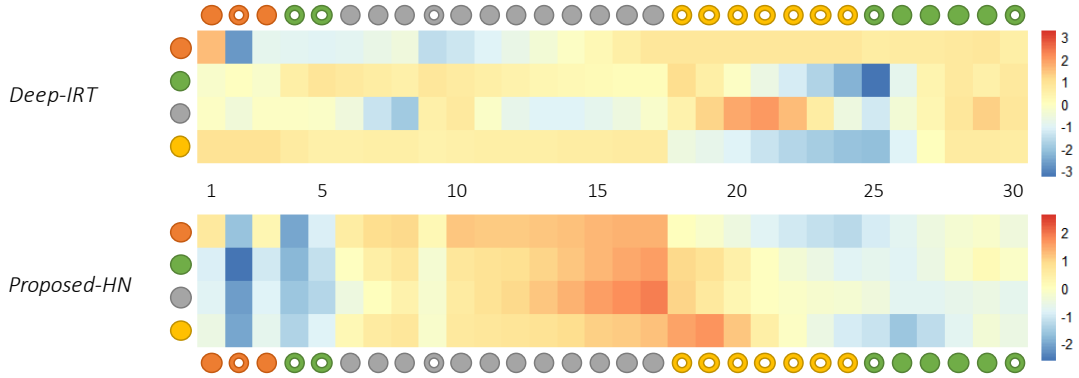


Figure 13: Example of a student ability transition from the ASSISTments2009 dataset. The skill inputs are classified respectively as ordering factions (orange), equation solving more than two steps (grey), equation solving two or fewer steps (green), finding percentages (yellow). The filled and the hollow circles respectively represent correct and incorrect responses.

skills. Therefore, when a student responds to an item, not only the corresponding skill ability but those for other skills change. Especially, because the skills of "equation solving more than two steps" (grey) and "equation solving two or few steps" (green) are similar, the ability changes of each skill also indicate a strong correlation. Consequently, the results demonstrate that the proposed method improves the interpretability of Deep-IRT.

It is noteworthy that the student's responses are not immediately reflected in the estimated ability change when the student provides a different response from the previous several continuous same responses. For example, the ability for "finding percents" (yellow) increases in items 18–19 despite incorrect responses because the Proposed-HN estimates the student's ability with the past responses. Then, the estimated ability values change slightly later when the student provides a different response from the previous several continuous same responses. In addition, the abilities for "ordering factions" (orange), "equation solving more than two steps" (grey), and "equation solving two or fewer steps" (green) decrease faster than that of "finding percents" (yellow) in items 17–18, which suggests that the 18-th item is related to untagged skills. The Proposed-HN estimates a student's ability by assessing not only the tagged skill but also the relation with other skills. However, when a teacher tags a skill to items inappropriately, the estimated ability of the skill does not reflect the response data accurately. In such cases, verifying the relation between each item and the corresponding skill is necessary.

7 Conclusions

This study proposed a novel IRT based on deep learning that models a student's response to an item by two independent redundant networks: a student network and an item network. Because of two independent redundant neural networks, the parameters of the proposed method can be interpreted to a considerable degree while maintaining high prediction accuracy.

First, we proposed a new IRT based on deep learning for test theory which has two independent redundant networks assuming that the ability is constant throughout the learning process. Although the standard IRT assume random sampling of students' abilities from a standard normal distribution, the proposed method can express actual students' abilities distributions flexibly because it does not follow a standard normal distribution. Therefore, it estimates students' abilities with high accuracy when the students are not sampled randomly from a single distribution or when there are no common items among the different tests. The two independent networks provide a more reliable and robust ability estimation for actual data than IRT does.

Next, we proposed a new IRT based on deep learning for knowledge tracing that estimates dynamic changes of student abilities in the learning process and predicts student performances. Furthermore, we improved the prediction accuracy of the proposed method by combining it with a novel hypernetwork. In the earlier memory updating component, the forgetting parameters, which control the degree of forgetting the past latent value memory, are optimized only from the current input data. That restriction might degrade the prediction accuracy of the proposed method because the value memory only insufficiently reflects the past learning information. The proposed hypernetwork can estimate the optimal forgetting parameters by balancing both the current input data and the past latent variables.

Experiments conducted with the benchmark datasets demonstrated that the proposed method improves both the ability parameter interpretability and the prediction accuracies of the earlier KT methods. Especially, results showed that the proposed method with the hypernetwork is effective for tasks with a long-term learning process. Experiments for the simulation dataset demonstrated that the proposed method provides stronger correlations with true parameters of TIRT than the earlier method does. Furthermore, the proposed method estimates the abilities robustly, even with aberrant values.

This study employed slightly redundant deep networks compared to earlier methods. In future work, we intend to use the proposed method to investigate the per-

formances of more-redundant and deeper networks. Additionally, we will try to optimize a hypernetwork to maximize the prediction accuracy for large datasets. Most recently, results of some studies have indicated that each item's characteristics differ according to their texts, although they require the same skill. To resolve this difficulty, they proposed KT methods to estimate the relation between the item's text content and the student's performance using the NLP technique or graph neural network [9, 11, 30, 34, 37, 38, 39]. As future work, we expect to incorporate the item's text content into the proposed method to improve the student performance prediction accuracy. Furthermore, deep-learning approaches for KT have been used for Computerized Adaptive Testing (CAT) [82, 83]. The main purpose of CAT is the measurement of the student ability in the personalized test for online education. Therefore, we infer that the proposed method might be effective for CAT because it can estimate the student's ability correctly.

Related journal papers

1. Emiko Tsutsumi, Ryo Kinoshita, Maomi Ueno, "Deep item response theory as a novel test theory based on deep learning," Electronics, Vol.10, Issue.9, no.1020 ,2021. (Section3.1)

2. Emiko Tsutsumi, Yiming Guo, Maomi Ueno,"DeepIRT with a Hypernetwork to optimize the degree of forgetting of past data," The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J106-D,No.02,Feb, 2023. (Section3.3)

Related conference papers

1. Emiko Tsutsumi, Ryo Kinoshita, Maomi Ueno, "Deep-IRT with independent student and item networks," in Proceedings of the 14th International Conference on Educational Data Mining (EDM), 2021. (Section3.2)

2. Emiko Tsutsumi, Yiming Guo, Maomi Ueno, "Deep knowledge tracing incorporating a hypernetwork with independent student and item networks," in Proceedings of the 15th International Conference on Educational Data Mining (EDM), 2022. (Section3.3)

Other papers

Journal papers

1. Maomi Ueno, Kazuma Fuchimoto, and Emiko Tsutsumi, "E-testing from artificial intelligence approach," Behaviormetrika,Vol.48,No.2,pp.409-424, 2021.(Invited paper)

2. Emiko Tsutsumi, Ryo Kinoshita, Maomi Ueno,"Deep-IRT with independent student and item networks," The Institute of Electronics, Information and Communication Engineers (IEICE), J104-D,No.7,pp.596-608, 2021.

3. Itsuki Aomi, Emiko Tsutsumi, Masaki Uto, Maomi Ueno,"Automated essay scoring model averaging by item response theory," The Institute of Electronics, Information and Communication Engineers (IEICE), J104-D,No.11,pp.784-795, 2021.

4. Emiko Tsutsumi, Ryo Kinoshita, Maomi Ueno,"Sliding window hidden markov item response theory for knowledge tracing," The Institute of Electronics, Information and Communication Engineers (IEICE), J103-D,No.12,pp.894-905, 2020.

5. Emiko Tsutsumi, Masaki Uto, Maomi Ueno,"Item response theory for dynamic assessment", The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J102-D,No.2,pp.79-92, 2019.

Conference paper

1. Itsuki Aomi, Emiko Tsutsumi, Masaki Uto, Maomi Ueno ,"Integration of automated essay scoring models using item response theory," in Proceedings of the International Conference on Artificial Intelligence in Education (AIED), 2021.

References

[1] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapt. Interact.*, vol. 4, no. 4, pp. 253–278, Dec 1995.

[2] Z. Pardos and N. Heffernan, "T.: Modeling individualization in a bayesian networks implementation of knowledge tracing," in *Proceedings of the 18th International Conference on User Modeling, Adaption, and Personalization*, 06 2010, pp. 255–266.

[3] Z. A. Pardos and N. T. Heffernan, "KT-IDEM: Introducing item difficulty to the knowledge tracing model," in *Proceedings of 19th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011)*, 01 2011, pp. 243–254.

[4] J. Lee and E. Brunskill, "The impact on individualizing student models on necessary practice opportunities," in *Proceedings of the Fifth International Conference on Educational Data Mining*, 01 2012, pp. 118–125.

[5] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *Artificial Intelligence in Education*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 171–180.

[6] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 505–513.

[7] C. Yeung, "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.

[8] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory network for knowledge tracing," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. International World Wide Web Conferences Steering Committee, 2017, pp. 765–774.

[9] 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, pp. 100–115, 06 2019.

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

[11] 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*, 2020, pp. 1205–1214.

[12] 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*, 2020.

[13] E. Tsutsumi, R. Kinoshita, and M. Ueno, "Deep-IRT with independent student and item networks," in *Proceedings of the 14th International Conference on Educational Data Mining (EDM)*, 2021.

[14] D. Agarwal, R. Baker, and A. Muraleedharan, "Dynamic knowledge tracing through data driven recency weights," in *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, 2020, pp. 725–729.

[15] S. Gowda, J. Rowe, R. Baker, M. Chi, and K. Koedinger, "Improving models of slipping, guessing, and moment-by-moment learning with estimates of skill difficulty." in *Proceedings of the Fourth International Conference on Educational Data Mining*, 01 2011, pp. 199–208.

[16] M. Khajah, Y. Huang, J. Gonzalez-Brenes, M. Mozer, and P. Brusilovsky, "Integrating knowledge tracing and item response theory: A tale of two frameworks," *Personalization Approaches in Learning Environments*, vol. 1181, pp. 5–17, 2014.

[17] J. Reye, "Student modelling based on belief networks," *International Journal of Artificial Intelligence in Education*, vol. 14, pp. 63–96, 2004.

[18] W. Hawkins, N. Heffernan, and R. Baker, "Learning bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities," in *Intelligent Tutoring Systems*. Springer Berlin Heidelberg, 06 2014, pp. 150–155.

[19] F. Baker and S. Kim, *Item Response Theory: Parameter Estimation Techniques, Second Edition*, ser. Statistics: A Series of Textbooks and Monographs. Taylor & Francis, 2004.

[20] C. Ekanadham and Y. Karklin, "T-SKIRT: Online estimation of student proficiency in an adaptive learning system," *CoRR*, vol. abs/1702.04282, 2017.

[21] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation," in *9th International Conference on Educational Data Mining*, vol. 1, 06 2016, pp. 539–544.

[22] F. Bartolucci, F. Pennoni, and G. Vittadini, "Assessment of school performance through a multilevel latent markov–rasch model," *Journal of Educational and Behavioral Statistics*, 09 2011.

[23] J. V. Dylan Molenaar, Daniel Oberski and P. D. Boeck, "Hidden markov item response theory models for responses and response times," *Multivariate Behavioral Research*, vol. 51, pp. 606–626, 2016.

[24] J. Gonzalez-Brenes, Y. Huang, and P. Brusilovsky, "General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge," in *Proceedings of the Seventh International Conference on Educational Data Mining*, 01 2014.

[25] J. H. Park, "Modeling preference changes via a hidden markov item response theory model," *In Handbook of Markov Chain Monte Carlo*, pp. 479–491, 2011.

[26] R. Weng and D. Coad, "Real-time bayesian parameter estimation for item response models," *Bayesian Analysis*, vol. 13, 12 2016.

[27] X. Wang, J. Berger, and D. Burdick, "Bayesian analysis of dynamic item response models in educational testing," *The Annals of Applied Statistics*, vol. 7, no. 1, pp. 126–153, 2013.

[28] H. Sepp and S. Jurgen, "Long short-term memory," *Neural Computation*, vol. 14, pp. 1735–1780, 1997.

[29] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, G. Fu, and G. Wang, "Concept-aware deep knowledge tracing and exercise recommendation in an online learning system," in *Proceedings of the International Conference on Educational Data Mining, EDM*, 2019.

[30] 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, no. 1, 2018, pp. 2435–2443.

[31] X. Sun, X. Zhao, Y. Ma, X. Yuan, F. He, and J. Feng, "Multi-behavior features based knowledge tracking using decision tree improved dkvmn," in *Proceedings of the ACM Turing Celebration Conference – China*. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3321408.3322847

[32] X. Xiong, S. Zhao, V. Inwegen, E. G., and J. E. Beck, "Going deeper with deep knowledge tracing," in *Proceedings of International Conference on Education Data Mining*, 2016.

[33] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang, "Neural cognitive diagnosis for intelligent education systems," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[34] S. Sonkar, A. E. Waters, A. S. Lan, P. J. Grimaldi, and R. Baraniuk, "qDKT: Question-centric deep knowledge tracing," *ArXiv*, vol. abs/2005.12442, 2020.

[35] Y. Zhou, Q. Liu, J. Wu, F. Wang, Z. Huang, W. Tong, H. Xiong, E. Chen, and J. Ma, "Modeling context-aware features for cognitive diagnosis in student learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '21. Association for Computing Machinery, 2021, pp. 2420–2428.

[36] W. Gao, Q. Liu, Z. Huang, Y. Yin, H. Bi, M.-C. Wang, J. Ma, S. Wang, and Y. Su, "RCD: Relation map driven cognitive diagnosis for intelligent education systems," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. Association for Computing Machinery, 2021, pp. 501–510.

[37] Y. Ma, P. Han, H. Qiao, C. Cui, Y. Yin, and D. Yu, "SPAKT: A self-supervised pre-training method for knowledge tracing," *IEEE Access*, vol. 10, pp. 72 145–72 154, 2022.

[38] Z. Wu, L. Huang, Q. Huang, C. Huang, and Y. Tang, "SGKT: Session graph-based knowledge tracing for student performance prediction," *Expert Systems with Applications*, vol. 206, p. 117681, 2022.

[39] Y. Luo, B. Xiao, H. Jiang, and J. Ma, "Heterogeneous graph based knowledge tracing," in *Proceedings of the 11th International Conference on Educational and Information Technology (ICEIT)*, 2022, pp. 226–231.

[40] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," in *Proceedings of International Conference on Education Data Mining*, 2019.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[42] H. He, G. Huang, and Y. Yuan, "Asymmetric valleys: Beyond sharp and flat local minima," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 2553–2564. [Online]. Available: http://papers.nips.cc/paper/8524-asymmetric-valleys-beyond-sharp-and-flat-local-minima.pdf

[43] A. Morcos, H. Yu, M. Paganini, and Y. Tian, "One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 4932–4942. [Online]. Available: http://papers.nips.cc/paper/8739-one-ticket-to-win-them-all-generalizing-lottery-ticket-initializations-across-datasets-and-optimizers.pdf

[44] V. Nagarajan and J. Z. Kolter, "Uniform convergence may be unable to explain generalization in deep learning," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 11 615–11 626. [Online]. Available: http://papers.nips.cc/paper/9336-uniform-convergence-may-be-unable-to-explain-generalization-in-deep-learning.pdf

[45] E. Tsutsumi, Y. Guo, and M. Ueno, "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.

[46] E. Tsutsumi, R. Kinoshita, and M. Ueno, "Deep item response theory as a novel test theory based on deep learning," *Electronics*, vol. 10, no. 9, 2021.

[47] F. Lord, "Applications of item response theory to practical testing problems," 1980.

[48] W. van der Linden, "Handbook of item response theory, volume two: Statistical tools," 2016.

[49] W. van der Linden, "Handbook of item response theory, volume three: Applications," 2016.

[50] S.-H. Joo, P. Lee, and S. Stark, "Evaluating anchor-item designs for concurrent calibration with the ggum," *Applied Psychological Measurement*, vol. 41, no. 2, pp. 83–96, 2017.

[51] H. David, D. Andrew, and V. L. Quoc, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[52] K. Stanley, D. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, pp. 185–212, 02 2009.

[53] G. Melis, K. Tomáš, and B. Phil, "Mogrifier LSTM," in *Proceedings of ICLR 2020*, 2020.

[54] B. Krause, L. Lu, I. Murray, and S. Renals, "Multiplicative LSTM for sequence modelling," *Workshop Track in ICLR*, 2017.

[55] Y. Wu, S. Zhang, Y. Zhang, Y. Bengio, and R. Salakhutdinov, "On multiplicative integration with recurrent neural networks," *Advances in Neural Information Processing Systems*, pp. 2856–2864, 2016.

[56] J. Koutník, F. Gomez, and J. Schmidhuber, "Evolving neural networks in compressed weight space," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 01 2010, pp. 619–626.

[57] C. Fernando, D. Banarse, M. Reynolds, F. Besse, D. Pfau, M. Jaderberg, M. Lanctot, and D. Wierstra, "Convolution by evolution: Differentiable pattern producing networks," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 07 2016, pp. 109–116.

[58] M. Moczulski, M. Denil, J. Appleyard, and N. Freitas, "ACDC: A structured efficient linear layer," in *ICLR*, 2016.

[59] M. Ueno and Y. Miyazawa, "Probability based scaffolding system with fading," in *Proceedings of Artificial Intelligence in Education – 17th International Conference, AIED*, 2015, pp. 237–246.

[60] M. Ueno and Y. Miyazawa, "IRT-based adaptive hints to scaffold learning in programming," *IEEE Transactions on Learning Technologies*, vol. 11, no. 4, pp. 415–428, Oct 2018.

[61] F. Lord and M. Novick, *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968.

[62] P. Chen, Y. Lu, V. Zheng, and Y. Pian, "Prerequisite-driven deep knowledge tracing," in *In IEEE International Conference on Data Mining, ICDM 2018*, 2018, pp. 39–48.

[63] C. K. Yeung and D.-Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proceedings of the Fifth ACM Conference on Learning @ Scale*, 2018, pp. 1–10.

[64] S. Cheng, Q. Liu, E. Chen, Z. Huang, Z. Huang, Y. Chen, H. Ma, and G. Hu, "DIRT: Deep learning enhanced item response theory for cognitive diagnosis," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2397–2400.

[65] Z. Wang, X. Feng, J. Tang, G. Huang, and Z. Liu, "Deep knowledge tracing with side information," in *Proceedings of the 20th International Conference on Artificial Intelligence in Education (AIED)*, 2019, pp. 303–308.

[66] G. Abdelrahman and Q. Wang, "Knowledge tracing with sequential key-value memory networks," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2019, pp. 175–184.

[67] B. Xu, S. Yan, and D. Yang, "BiRNN-DKT: Transfer bi-directional LSTM RNN for knowledge tracing," in *Web Information Systems and Applications*, W. Ni, X. Wang, W. Song, and Y. Li, Eds. Cham: Springer International Publishing, 2019, pp. 22–27.

[68] X. Liangbei and D. Mark, "Dynamic knowledge embedding and tracing," in *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, 2020, pp. 524–530.

[69] Y. Lu, D. Wang, Q. Meng, and P. Chen, "Towards interpretable deep learning models for knowledge tracing," in *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, 2020, pp. 185–190.

[70] H. Tong, Y. Zhou, and Z. Wang, "Exercise hierarchical feature enhanced knowledge tracing," in *Proceedings of the 13th International Conference on Educational Data Mining, EDM*, 2020, pp. 324–328.

[71] L. Donghua, J. Yanming, Z. Jian, W. Wufeng, and X. Ning, "Deep knowledge tracing based on bayesian neural network," in *Advances in Intelligent Systems and Interactive Applications*, F. Xhafa, S. Patnaik, and M. Tavana, Eds. Cham: Springer International Publishing, 2020, pp. 29–37.

[72] R. Georg, "Probabilistic models for some intelligence and attainment tests," *MESA Press*, 1993.

[73] P. Songmuang and M. Ueno, "Bees algorithm for construction of multiple test forms in e-testing," *IEEE Transactions on Learning Technologies*, vol. 4, pp. 209–221, 07 2011.

[74] T. Ishii, P. Songmuang, and M. Ueno, "Maximum clique algorithm for uniform test forms assembly," in *The 16th International Conference on Artificial Intelligence in Education*, vol. 7926, 07 2013, pp. 451–462.

[75] T. Ishii, P. Songmuang, and M. Ueno, "Maximum clique algorithm and its approximation for uniform test form assembly," *IEEE Transactions on Learning Technologies*, vol. 7, pp. 83–95, 01 2014.

[76] T. Ishii and M. Ueno, "Clique algorithm to minimize item exposure for uniform test forms assembly," in *International Conference on Artificial Intelligence in Education*, 06 2015, pp. 638–641.

[77] T. Ishii and M. Ueno, "Algorithm for uniform test assembly using a maximum clique problem and integer programming," in *Artificial Intelligence in Education. Springer International Publishing*, 06 2017, pp. 102–112.

[78] Y. Lin, Y.-S. Jiang, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, "A discrete multi-objective particle swarm optimizer for automated assembly of parallel cognitive diagnosis tests," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–14, 06 2018.

[79] J.-J. Vie, F. Popineau, E. Bruillard, and Y. Bourda, "Automated test assembly for handling learner cold-start in large-scale assessments," *International Journal of Artificial Intelligence in Education*, vol. 28, 02 2018.

[80] W. Linden and B. Jiang, "A shadow-test approach to adaptive item calibration," *Psychometrika*, vol. 85, 06 2020.

[81] H. Ren, S. Choi, and W. Linden, "Bayesian adaptive testing with polytomous items," *Behaviormetrika*, vol. 47, 05 2020.

[82] H. Bi, H. Ma, Z. Huang, Y. Yin, Q. Liu, E. Chen, Y. Su, and S. Wang, "Quality meets diversity: A model-agnostic framework for computerized adaptive testing," in *Proceeding of the 2020 IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 42–51.

[83] Y. Zhuang, Q. Liu, Z. Huang, Z. Li, S. Shen, and H. Ma, "Fully adaptive framework: Neural computerized adaptive testing for online education," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, Jun. 2022, pp. 4734–4742. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20399

[84] W. Shen, X. Wang, X. Bai, and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3982–3991.

[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[86] M. Reckase, "Multidimensional item response theory models," *Springer*, 2009.

[87] S. Kilmen and N. Demirtasli, "Comparison of test equating methods based on item response theory according to the sample size and ability distribution," *Procedia – Social and Behavioral Sciences*, vol. 46, pp. 130–134, 2012, 4th World Conference on Educational Sciences (WCES-2012) 02-05 February 2012 Barcelona, Spain.

[88] I. Uysal and S. Kilmen, "Comparison of item response theory test equating methods for mixed format tests," *International Online Journal of Educational Sciences*, vol. 2016, 06 2016.

[89] W. van der Linden and M. D. Barrett, "Linking item response model parameters," *Psychometrika*, vol. 81, no. 3, pp. 650–673, Sep 2016.

[90] M. Ueno, "Animated agent to maintain learner's attention in e-learning," in *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*. Washington, DC, USA: Association for the Advancement of Computing in Education (AACE), 2004, pp. 194–201.

[91] M. Ueno, "Data mining and text mining technologies for collaborative learning in an ILMS "Samurai"." in *Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT 2004*, 01 2004, pp. 1052–1053.

[92] M. Ueno, "Intelligent LMS with an agent that learns from log data," in *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005*, G. Richards, Ed. Vancouver, Canada: Association for the Advancement of Computing in Education (AACE), October 2005, pp. 3169–3176.

[93] A. C. George, A. Robitzsch, T. Kiefer, J. Groß, and A. Ünlü, "The R package CDM for cognitive diagnosis models," *Journal of Statistical Software, Articles*, vol. 74, no. 2, pp. 1–24, 2016.

[94] H.-S. Chang, H.-J. Hsu, and K.-T. Chen, "Modeling exercise relationships in e-learning: A unified approach," in *EDM*, 2015.

[95] Z. Wang, A. Lamb, E. Saveliev, P. Cameron, Y. Zaykov, J. M. Hernández-Lobato, R. E. Turner, R. G. Baraniuk, C. Barton, S. P. Jones, S. Woodhead, and C. Zhang, "Instructions and guide for diagnostic questions: The NeurIPS 2020 education challenge," *arXiv preprint arXiv:2007.12061*, 2020.