# 修 士 論 文 の 和 文 要 旨

| | | | |
|---|---|---|---|
| 研究科・専攻 | 大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程 | | |
| 氏　　名 | GUO YIMING | 学籍番号 | 2131175 |
| 論 文 題 目 | Adversarial training for Deep-IRT with a hypernetwork to optimize the degree of past data<br>（学習データの忘却を最適化するハイパーネットワークを用いたDeep-IRTの敵対的学習） | | |


要　旨

近年，人工知能分野において，機械学習を用いてユーザのオンライン上での履歴から能力および各スキルの習熟度を推定し，未知の反応予測を行うKnowledge Tracing(KT)が盛んに研究されている。特に，最新のKT手法として深層学習を基づいた様々な手法が提案されている。その中で、深層学習手法と項目反応理論(Item Response Theory; IRT)を組み合わせるDeep-IRTは高い反応予測精度とパラメータ解釈性を備えているため注目されている。しかし、既存のDeep-IRTでは能力値を推定する際に最新の履歴データのみを用いるため、過去のデータを予測精度に十分に反映できていない可能性がある。本研究では、Deep-IRTに新たなハイパーネットワークを組み合わせることで、ユーザの過去の履歴データの忘却を最適化しながら反応予測を行う。さらに、パラメータ学習に敵対的学習を導入することで、過学習の問題を改善し反応予測精度をより向上させる。評価実験では提案手法と既存手法の反応予測精度を比較することで、提案手法の有効性を示した。さらに、IRTモデルを用いてシミュレーションデータを作成し、提案手法と既存手法によって推定された能力パラメータと真の能力パラメータの相関係数を比較することで解釈性も向上させたことを示した。

# Adversarial training for Deep-IRT with a hypernetwork to optimize the degree of past data

The University of Electro-Communications

2131175

GUO YIMING

Advisors:

Prof. Maomi UENO

Assoc. Prof. Masaki UTO

guo_yzmzng@ai.lab.uec.ac.jp

# Contents

**Abstract** ..... 4

**Chapter 1: Introduction** ..... 5

**Chapter 2: Related works** ..... 9

**1. Item Response Theory** ..... 9
**2. Dynamic Key-Value Memory Network** ..... 10
**3. Deep-IRT** ..... 12
**4. Context-Aware Attentive Knowledge Tracing** ..... 14

**Chapter 3: The proposed method** ..... 16

**1. Hypernetwork** ..... 17
**2. Modified memory updating component** ..... 18
**3. Adversarial training** ..... 20

**Chapter 4: Experiments** ..... 22

**1. Hyper-parameter selection** ..... 23
**2. Results** ..... 25
**3. Analysis of interpretable parameters** ..... 27

**Chapter 5: Conclusions** ..... 30

**Acknowledgement** ..... 31

**References** ..... 32

# Tables and Figures

Figure 1: Network architecture of DKVMN and Deep-IRT.....10

Figure 2: Network architecture of Deep-IRT(Tsutsumi et al. 2021)..... 13

Figure 3: Memory updating component of the proposed DeepIRT (Tsutsumi et al. 2021) with hypernetwork ..... 17

Figure 4: Example of a learner transition from the ASSISTments2009 dataset. The comparison between Deep-IRT (Tsutsumi et al. 2021) and the proposed method.....28

Table 1. Prediction Accuracy of Alleviating Over-fitting Approaches..... 21

Table 2. Summary of Benchmark Datasets.....22

Table 3. Prediction Accuracy and Hyper-Parameters r..... 23

Table 4. Prediction Accuracy of Student's Performance with Only Skill Inputs ..... 24

Table 5. Prediction Accuracy of Student's Performance with both Item and Skill Inputs ..... 25

Table 6. Correlation Coefficient's of The Estimated Abilities .....26

# Abstract

Knowledge Tracing (KT), which can assess learners' knowledge levels, has traditionally been performed manually. With the advancements in artificial intelligence (AI), deep learning approaches have gained extensive attention and demonstrated superior performance. Furthermore, the latest Deep-IRT model has been reported to achieve higher accuracy and provide a better interpretation of parameters than the previous KT methods did. Nevertheless, there is still room for improvement in the memorization process, especially for tasks involving longer sequences of learner responses. In the field of natural language processing (NLP), the Mogrifier as a type of the hypernetwork has shown great performance in optimizing the memorization process, particularly for tasks with longer sequences. To address this issue, this paper proposes a new memory updating component with a hypernetwork to optimize the balance between the current input and the past data. Furthermore, the model suffers from over-fitting, we introduce adversarial training (AT) for the proposed method. Experimental results demonstrate that the proposed method improves the prediction accuracy and the interpretability of the learners' ability compared to existing KT methods.



Keywords: Knowledge Tracing, deep learning, long-term learning history, hypernetwork

# Chapter 1: Introduction

Recently, with the development of online learning systems, Knowledge Tracing (KT) has gained extensive attention for its feature to accurately estimate the degree of a learner's skills. By accurately assessing learners' skills, educators can receive better feedback and design personalized study plan for learners with varying knowledge levels [1,2,3,4,5,6,7,8,9,10]. Learners can also benefit from the analyses to understand their level of skills.

As a result, researchers in the field of Artificial Intelligence (AI) have proposed various kind of Knowledge Tracing (KT) methods. The KT methods can be generally divided into two categories: probabilistic approaches and deep-learning approaches. Bayesian Knowledge Tracing (BKT) is a famous probabilistic model for KT [1]. It employs a Hidden Markov Model (HMM) to track the process of a learner's ability change in the learning process [1]. BKT estimates whether a learner has mastered a skill or not and expresses it in binary proficiency parameters. Based on the parameters, the BKT predicts the learners' answers to unknown items. Many researchers have proposed additional BKT models to improve interpretability. However, the earlier BKT models only address simple discrete values. As a result, BKT lacks the flexibility to capture changes in ability and cannot analyze tasks with multiple skills. Recently, Item Response Theory (IRT) has been used for KT [25]. IRT predicts the positive answer probability based on the difficulty of items and the ability of learners. However, traditional IRT models cannot track changes in learners' abilities. To address the issue, some researchers proposed novel IRTs combining with HMM [26]. But another problem remains that probabilistic methods are unable to capture the correlations between different skills and reflect them on the learners' abilities.

To overcome the limitations, deep learning approaches were proposed. The powerful feature extraction capability enables model to better capture the potential connections between various skills. As a result, deep learning methods can achieve higher accuracy in predicting learners' responses.

Deep Knowledge Tracing (DKT) [6], as the first deep learning method in the field of KT, employs a Long Short Term Memory (LSTM) [27] to track a learner's study process. DKT predicts the performance based on the study history and uses the hidden state to memorize the learner's learning history data. With the powerful feature

extraction capability of deep learning, DKT addresses the issue of assuming random sampling learners' abilities for each skill, thus improved prediction accuracy compared to probabilistic methods. However, the hidden states include a summary of the learning history data in LSTM, but individual skills are not treated separately. As a result, DKT does not explicitly account for the learner's mastery in each skill.

Several studies have proposed various extensions to improve the accuracy of DKT [28,29,30]. The Dynamic Key-Value Memory Network (DKVMN) [8] is well known as one of the most representative extensions. DKVMN first introduces the key-value structure to establish correlations between skills [8]. The key matrix, combined with an attention mechanism, captures the correlations between skills. Then the skills will be transformed into the latent Knowledge Components (KCs) which represent the correlation relationships of skills. The value matrix stores learners' knowledge statuses to the KCs and is updated at each time point. Furthermore, DKVMN predicts the learner's performance based on the current item in addition to the learning history [8]. Although DKVMN brings higher prediction accuracy, it lacks the interpretable parameters that probabilistic methods have. To address the limitation, Deep-IRT was proposed by combining IRT and DKVMN [7]. Deep-IRT can estimate learners' abilities and items' difficulties by information stored in the latent parameters. Deep-IRT improves interpretability while maintaining the high prediction accuracy of DKVMN. However, Deep-IRT assumes that items sharing the same skill are identical. This approach might not be effective when items sharing the same skill exhibit significant differences, thus preventing the model from accurately measuring learners' abilities.

Most of the previous methods were based on Recurrent Neural Networks (RNNs), which simulate the learning process of learners. With the development of AI in the field of Natural Language Processing (NLP), the Transformer introduced the self-attention mechanism which changed the landscape for using RNN models [17]. The self-attention mechanism was proved to have powerful feature extraction capabilities in many different tasks. Self-attentive Knowledge Tracing (SAKT) [16] was the first model applied Transformer to KT. SAKT predicts the learners' performance by measuring the correlation of the responses and items in their study histories. However, due to the difference between KT and NLP tasks, earlier learning histories fail to influence learners' performance regularly in KT. As a result, SAKT fails to show a high accuracy. Gosh et al. (2020) proposed Context-aware Attentive Knowledge Tracing (AKT) [9],

which introduced a modified self-attention mechanism. In calculating the correlation between skills in self-attention, they incorporated an exponential function to assign less importance to items which are further away from the current time point. The consideration of the prior information allows the model better captures changing patterns of learners' performance. However, the modified self-attention algorithm recalculates correlations between each skill at each time point. Additionally, the results heavily depend on the nearby learning history data due to the exponential function, AKT fails to track learners' knowledge transitions.

To improve the prediction accuracy and the interpretability of learners' knowledge statuses, Tsutsumi et al (2021) proposed Deep-IRT with Independent Student and Item Networks [10]. The item network plays a crucial role in analyzing both the item and its associated skill [10]. The independent item network, with an adjustable number of parameters, enables more accurate estimation of difficulty [10]. Moreover, the model can distinguish between different items that share the same skill [10]. It allows IRT module more accurately captures learners' abilities and the future performance [10,35]. However, issues remained in the memory updating component inherited from DKVMN model. When the parameters storing the learners' knowledge statuses are updated, the variables controlling the update process are solely optimized based on the latest response data. It results in knowledge statuses' updates are consistent for learners with the same response, regardless of their knowledge levels. Additionally, the model treats previous and current responses in learners' study histories as separate entities without establishing connections between them. It poses a risk of losing the relevant information for updates. To address these issues, this paper proposes the incorporation of a hypernetwork to optimize the update process [18].

In the field of NLP , the concepts of hypernetwork were introduced to optimize hidden parameters of RNN models [18]. Hypernetworks enable models to effectively capture information in sentences during time transitions and avoid information loss from the input data [18,20]. Similarly, KT tasks share features with NLP tasks. Therefore, we propose to incorporate a hypernetwork to balance the current and the past data [34]. The hypernetwork optimizes both the forgetting parameters and the variables that store learners' knowledge level in the learning process [34]. It enables the memory updating component to better estimate the degree of forgetting past data [34]. Additionally, we design a modified memory updating component to effectively utilize

the optimized data and reduce the loss of the relevant information [34]. We present our research in the paper "Deepirt with a hypernetwork to optimize the degree of forgetting of past data." [34]. However, due to the incorporation of more parameters from the modified component and the small scale of datasets in KT, the proposed method suffers from over-fitting. Adversarial training (AT) was initially proposed in the field of computer vision (CV) and is widely used in various fields to alleviate the over-fitting [22,23,24]. We introduce the AT to address the issue. To validate the effectiveness of the proposed method, we provide experimental results to compare the performance with previous KT methods. The results demonstrate the proposed method outperforms previous KT methods in terms of prediction accuracy and interpretability, particularly for tasks with long-term learning histories [20].

# Chapter 2: Related works

## 2.1 Item Response Theory

Item Response Theory (IRT) is a logistic regression model. IRT models are widely used in educational settings, particularly in the adaptive testing, to infer learners' abilities and provide items that match their knowledge levels [25]. IRT models assume abilities of learners remain constant during the exam. Among various IRT models, the two-parameter logistic model (2PLM) is a well-known and widely used model [25]. In the 2PLM, $\boldsymbol{u_{ij}}$ denotes the response of learner $\boldsymbol{i}$ to item $\boldsymbol{j} \in (\mathbf{1},\ldots, \mathbf{n})$ as below:

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

The $\boldsymbol{P_j(\theta_i)}$ represents the probability of the learner $\boldsymbol{i}$, with the ability $\boldsymbol{\theta}$, can answer the item $\boldsymbol{j}$ with the difficulty $\boldsymbol{\beta_j}$ correctly [25]. The probability is determined by learners' abilities and difficulties of items. The function is shown below:

$$\boldsymbol{P_j(\theta_i)} = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{exp}(-\boldsymbol{\alpha_j}(\boldsymbol{\theta_i}-\boldsymbol{\beta_j}))} . \quad (2)$$

The $\boldsymbol{\alpha_j}$ represents the discrimination parameter of item $\boldsymbol{j}$ in IRT model, which expresses the discriminatory power for learners' abilities. In the standard IRT model, the learner's ability is assumed to be constant during the learning process. However, some researchers proposed the combination of the IRT model and HMMs to capture changes in the learner's abilities over time [26]. But for the IRT model, ability parameters are single-dimensional. As a result, the model can not capture correlations between different skills and reflect correlations on learners' abilities.

## 2.2 Dynamic Key-Value Memory Networks

DKT is the first deep learning approach of KT. DKT employs LSTM to capture a learner's knowledge status and simulate his learning process [6]. However, DKT summarizes a learner's abilities of various skills in the hidden state without distinct treatment [8]. Moreover, DKT employs LSTM without adapting it to characteristics of KT. Consequently, DKT fails to achieve both accurate predictions and parameter interpretability.

DKVMN is an extension of DKT that aims to improve the interpretability in terms of learners' abilities. The structure of the model is shown in Figure 1. The model first introduces the key-value structure. The key matrix $\boldsymbol{M_k} \in \boldsymbol{R^{N*d_k}}$ is a storage of correlations of skills and employs an attention mechanism to transform the actual skills into the attention weight $\boldsymbol{w_{tl}}$ of latent KCs [8].

$$\boldsymbol{w_{tl}} = \boldsymbol{Softmax}(\boldsymbol{M_l^k k_t}). \tag{3}$$

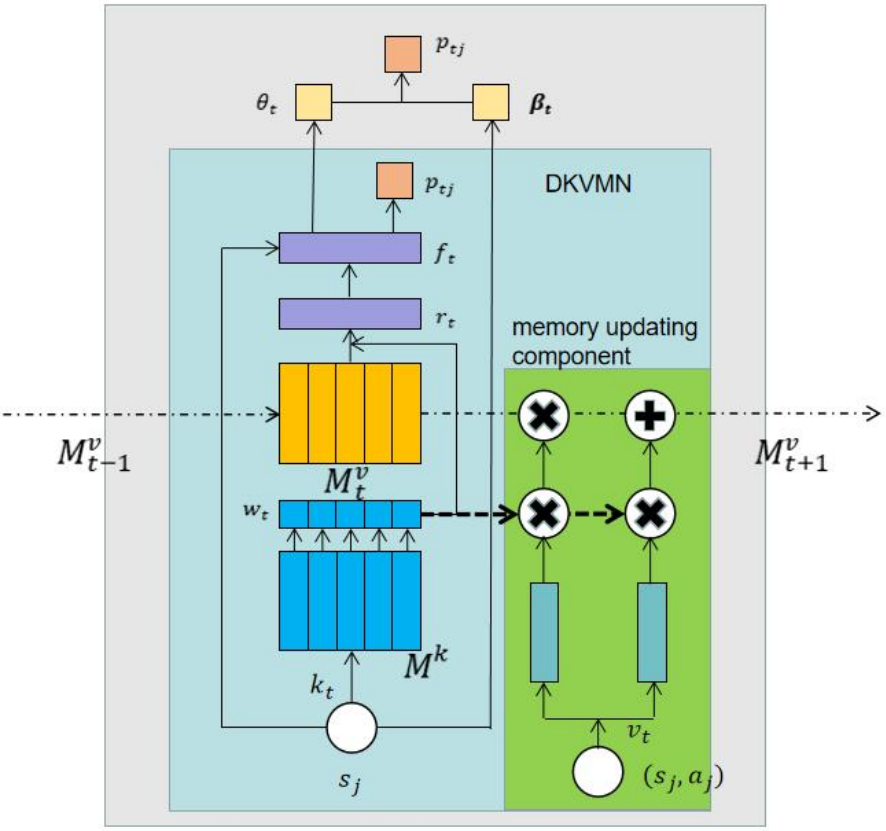


Figure 1: network architecture of DKVMN and Deep-IRT

Where $M_l^k$ represents the ***l***-th row vector and $w_{tl}$ shows the attention weight of item ***j*** on the latent KC ***l*** which expresses how strong the correlation between them is. And the ***t*** means a learner's response to item ***j*** at time point ***t***. $k_t$ is the embedded vector of item ***j***.

Furthermore, learners' knowledge statuses of KCs are stored in the value matrix $M_t^v \in R^{N*d_v}$. $M_t^v$ will be updated at each time point based on $w_t$ and $M_t^v$ to calculate vector $r_t$ which contains the sum of the ability on each KC. $M_{tl}^v$ represents the ***l-th*** row vector of matrix. $^\intercal$ denotes the transposition of matrix.

$$r_t = \sum_{i=1}^{N} w_{tl} M_{tl}^{v\intercal} . \quad (4)$$

One key difference from DKT is that DKVMN predicts the performance of learners based on both their knowledge statuses $M_t^v$ and $k_t$. It is opposed to DKT solely relying on the learner' knowledge level. The approach provides the additional information and improve the performance [6,8]. The function is shown below:

$$f_t = tanh(W^f[r_t \cdot k_t] + b^f), \text{ and} \quad (5)$$

$$P_{tj} = \sigma(W^P f_t + b^P). \quad (6)$$

**[·]** represents the concatenation of vectors, $\sigma(\cdot)$ is the sigmoid activation and $tanh$ is the hyperbolic tangent activation. $W$ is the weight parameter and $b$ is the bias parameter of layer. $f_t$ is the intermediate variable and $P_{tj}$ is the probability of the learner's response to the item ***j*** at time point ***t*** correctly.

The research also proposed an RNN structure for updating learners' knowledge statuses, with the reference to LSTM [27,31]. At time point ***t***, $M_t^v$ is updated with the embedded vector $v_t$ of input data $(s_j, a_j)$, which represents the learner's answer $a \in$ **{0,1}** towards the item with skill ***j***. The parameters $\mathbf{a}_t$ and $\mathbf{e}_t$ control the updating of $M_t^v$, $\mathbf{a}_t$ donates the part of the information that needs to be added to the current knowledge status and the $\mathbf{e}_t$ indicates the degree of the previous knowledge status should be forgotten. They are calculated with $w_{tl}$ to get the ***l-th*** latent KC's updating value.

$$\mathbf{e}_t = \sigma(W^e v_t + b^e) \ , \tag{7}$$

$$\mathbf{a}_t = tanh(W^a v_t + b^a) \ , \tag{8}$$

$$\widetilde{M}^v_{t,l} = M^v_{t,l} \cdot (\mathbf{1} - w_{tl} e_t)^{\mathsf{T}} \ \text{, and} \tag{9}$$

$$M^v_{t+1,l} = \widetilde{M}^v_{t,l} + w_{tl} \mathbf{a}_t \ . \tag{10}$$

The $M^v_{t+1,l}$ represents the value matrix will be at the time point ***t+1*** and $\widetilde{M}^v_{t,l}$ is the intermediate variable.

## 2.3 Deep-IRT

To improve the interpretability of DKVMN, Yeung et al (2019) proposed Deep-IRT as the combination of DKVMN and IRT [6,7,25]. Deep-IRT provides interpretable parameters in IRT model while maintaining the high prediction accuracy of DKVMN. The structure of Deep-IRT is shown in Figure 2. Deep-IRT adds two new hidden layers to estimate items' difficulties and learners' abilities as below:

$$\theta_{t,j} = tanh(W_\theta f_t + b_\theta), \text{ and} \tag{11}$$

$$\beta_{t,j} = tanh(W_\beta k_t + b_\beta). \tag{12}$$

Where the $\theta_{t,j}$ represents that at time point ***t*** the learner's ability to the item with skill ***j*** and the $\beta_{t,j}$ represents the difficulty of the item. With the hyperbolic tangent activation the value of them will be restricted to between **(-1,1) [7]**. The prediction will be modified to the form of IRT with both of them as below:

$$P_{tj} = \sigma(3 * \theta_{t,j} + \beta_{t,j}) \ . \tag{13}$$

As the $\sigma(\max(\theta_{t,j}) - \min(\beta_{t,j})) = \sigma(2) \approx 0.881$, $\theta_{t,j}$ multiplies with **3** can increase the accuracy of predictions [7]. However, despite the improvement in the interpretability, Deep-IRT assumes that items sharing the same skill are identical, which can decrease the accuracy of the model's IRT parameters [6,7].

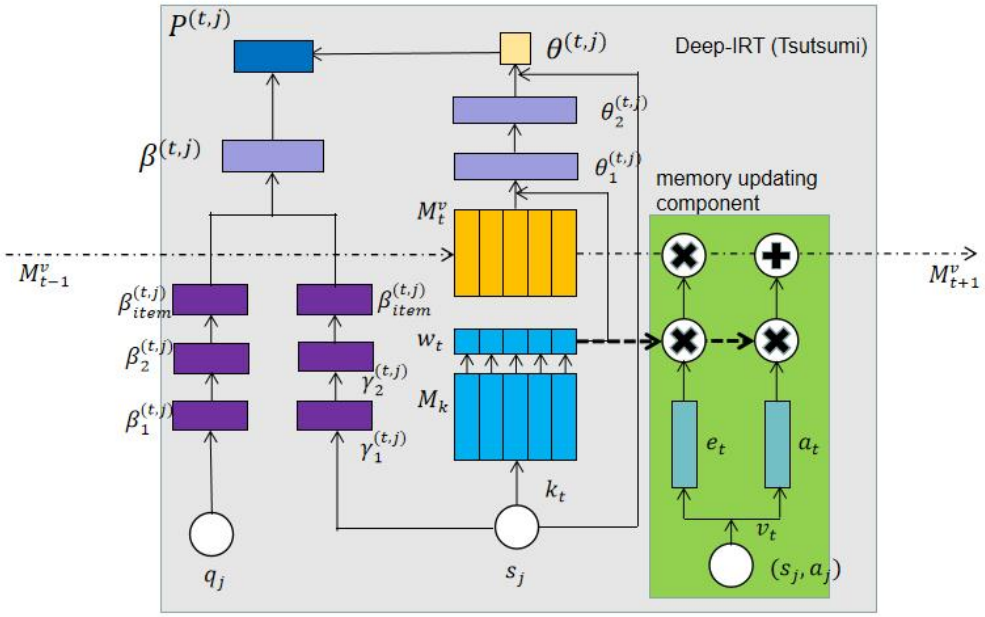


Figure 2: network architecture of Deep-IRT(***Tsutsumi et al. 2021***)

Therefore, Tsutsumi et al.(2021) proposed a Deep-IRT with independent student and item networks. The independent student and item networks help model better capture the interpretable value of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ [19,34]. The structure of Deep-IRT (***Tsutsumi et al. 2021***) is shown in Figure 2. In Deep-IRT (***Tsutsumi et al. 2021***), the presumption of $\boldsymbol{\theta}$ will no longer depend on the features of items, allowing the model captures changes in learners' abilities on the multi-dimensional KCs independently [19,34]. The addition of multiple layers in the independent item network allows for more precise feature extraction from items. It also enables the model to better distinguish different items that share the same skill. Deep-IRT(***Tsutsumi et al. 2021***) demonstrates high prediction accuracy equivalently and provides more accurate estimatation of interpretable parameters.

## 2.4 Context-Aware Attentive Knowledge Tracing

AKT demonstrates the best performance of predicting learners' responses among previous methods [9]. AKT deploys a Transformer backbone model, which is widely used in the field of NLP. The self-attention mechanism of Transformer demonstrates the superior performance on the prediction accuracy in various tasks. Similar to the ordinary Transformer model, AKT has a dual structure of self-attention mechanisms: the encoder and the decoder [17]. The function of self-attention is shown below:

$$\boldsymbol{\alpha_{t,r}} = \frac{\boldsymbol{exp(f_{t,r})}}{\sum_{r'} \boldsymbol{exp(f_{t,r'})}} , \tag{14}$$

$$\boldsymbol{f_{t,r}} = \begin{cases} \frac{\boldsymbol{q_t^\top k_r}}{\sqrt{\boldsymbol{D_k}}} , & (Encoder) \\ \frac{\boldsymbol{exp(-\eta d(t,r)) \cdot q_t^\top k_r}}{\sqrt{\boldsymbol{D_k}}} , & (Decoder) \end{cases} . \tag{15}$$

Where $\boldsymbol{q_t} \in \boldsymbol{R^{D_k}}$ is the query matrix of self-attention that represents learners' response data from time point ***1*** to ***t***. $\boldsymbol{k_r} \in \boldsymbol{R^{D_k}}$ is the key matrix of self-attention. $\boldsymbol{k_r}$ also represents the information of the response data and self-attention calculates items' correlation by multiplying $\boldsymbol{q_t}$ and $\boldsymbol{k_r}$. $\boldsymbol{D_k}$ is the dim size of matrix. $\boldsymbol{\alpha_{t,r}}$ is the attention between the data at time point ***t*** and ***r***.

The encoder in AKT calculates the attention of the current item to each previous data [9]. Only the data with high relevance to the current data will be extracted. In the research field of NLP, words with the valuable information could be present anywhere in a sentence. The self-attention mechanism can correctly extract them regardless of their positions. However, in the case of learners' study process, premature learning histories fail to influence learners' performance regularly [9]. Therefore, AKT incorporates the decoder with a forgetting function, which employs an exponential function, to reduce the importance of items that are further away from the current time point [9]. The structures of the encoder and the decoder are shown below:

$$\boldsymbol{d(t,r)} \;=\; |\boldsymbol{t-r}| \sum_{\boldsymbol{t'}}^{\boldsymbol{t}} \frac{\frac{\boldsymbol{q_t}^{\top}\boldsymbol{k_r}}{\sqrt{\boldsymbol{D_k}}}}{\sum_{\boldsymbol{1\le r'\le t'}} \frac{\boldsymbol{q_t}^{\top}\boldsymbol{k_{r'}}}{\sqrt{\boldsymbol{D_k}}}} \;. \tag{16}$$

As a result, in the decoder, the attention $\boldsymbol{\alpha_{t,r}}$ between the current and each past responses will be reduced by the elapsed time $|\boldsymbol{t-r}|$ to simulate learners' forgetting process. The algorithm enables AKT to incorporate the prior information into the model and learn about the particular dataset distribution, leading to a significant improvement in prediction accuracy. Moreover, the attention $\boldsymbol{\alpha_{t,r}}$ serves as a good representation of the correlation between the each item, thus improving the model's interpretability [9,17]. However, the results heavily depend on the nearby historical information due to the exponential function. As a result, The approach results in significant changes in predictions at each time point and hinders the model's ability to estimate the learners' knowledge levels accurately.

# Chapter 3: The proposed method

Tsutsumi et al. (2021) proposed a new Deep-IRT with two independent networks, the item network and the student network. The method improves the interpretation of estimating difficulty and ability parameters compared to previous methods. However, there is still room for the improvement in the memory updating component. The previous structure, inherited from DKVMN, encounters following challenges:

1. The memory updating component calculates the degree of updating based on connections between the current response and learning histories. However, the current and the past response data are treated separately until (9). Therefore, the model insufficiently captures the correlations between responses at each time point. The issue leads to the loss of the valuable information for updates of the value matrix [20].

2. The parameters $\mathbf{a}_t$ and $\mathbf{e}_t$, which update the value matrix $\boldsymbol{M}_t^v$, are solely calculated based on the current response. Consequently, learners with the same response will receive the similar degree of updating. However, it is evident that learners with varying knowledge levels could exhibit different changes in abilities even if they have the same response.

These issues lead to the inaccurate estimation of learners' abilities and have a negative impact on the the accuracy of predictions. To address above issues, we propose a modified memory updating component with a hypernetwork to optimize the update process. Recently, the concept of hypernetworks has been proposed as an extension of RNN models. On the basis, Melis et al. (2020) introduced the "Mogrifier component" as a type of the hypernetwork [18,20]. The Mogrifier component enables interaction between the input data and the hidden state before entering the gate unit of LSTM [18,20]. As a result, it can reduce the loss of the relevant information for updates and improving predicting accuracy, particularly for datasets with longer sequences [20].

In this paper, we propose the incorporation of a hypernetwork to help Deep-IRT in optimizing the updating degree of past data. We also modify the memory updating component to incorporate the capabilities of the hypernetwork.

## 3.1 Hypernetwork

In the memory updating component inherited from DKVMN, the parameters $\mathbf{a_t}$ and $\mathbf{e_t}$ that control the updating degree of $\boldsymbol{M_t^v}$ are solely optimized based on the current response data $\boldsymbol{v_t}$. It hinders the component from accurately capturing the connection between $\boldsymbol{v_t}$ and past responses ($\boldsymbol{v_1}$,...,$\boldsymbol{v_{t-1}}$) which are stored in $\boldsymbol{M_t^v}$. To address the issue, we propose the incorporation of a hypernetwork [18]. The hypernetwork enables the memory updating component to optimize the balance between $\boldsymbol{v_t}$ and $\boldsymbol{M_t^v}$. As a result, it helps the model extract more relevant information for updates [20,34].

Figure 3 shows the structure of the combination of the hypernetwork and the memory updating component. The input data of the hypernetwork $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{v}}$ is calculated as below:

$$\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{v}} = \begin{cases} \boldsymbol{M_t^v}, & (\boldsymbol{\lambda = 0}) \\ \boldsymbol{\sigma}(\boldsymbol{W}[\boldsymbol{M_t^v}, \boldsymbol{M_{t-1}^v}, \ldots, \boldsymbol{M_{t-\lambda}^v}] + \boldsymbol{b}), & (\boldsymbol{otherwise}) \end{cases}. \quad (17)$$

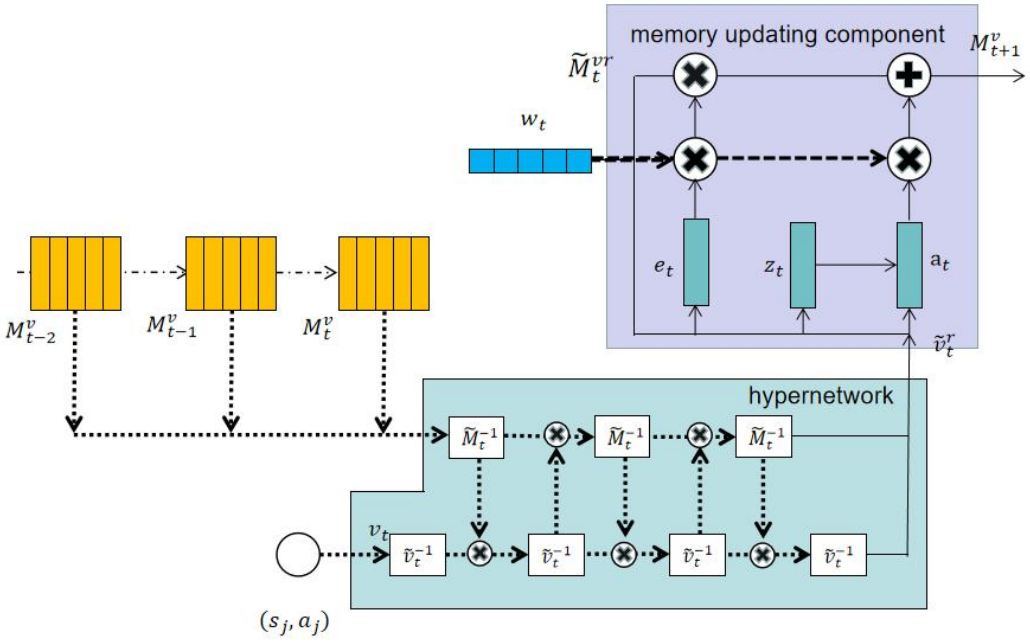


Figure 3: Memory updating component of the proposed DeepIRT (***Tsutsumi et al. 2021***) with hypernetwork.

We propose to incorporate the past knowledge statuses into the hypernetwork. It helps the hypernetwork include the information that might have been forgotten at previous time points but is relevant to the current data. The presence of more relevant data optimizes updating parameters for the current response data $\boldsymbol{v_t}$. The $\boldsymbol{\lambda} \in \{\mathbf{0}, \mathbf{1}, \ldots, \mathbf{t}\}$ is the hyper-parameter which determines the number of previous value matrices that should be included. **[]** represents the concatenate of matrices. In hypernetwork, $\boldsymbol{v_t}$ and $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{v}}$ are optimized as below:

$$\widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{r}} = \boldsymbol{\delta_1} * \boldsymbol{\sigma}(\boldsymbol{W}^{\boldsymbol{v}} \widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{vr-1}}) \otimes \boldsymbol{v}_{\boldsymbol{t}}^{\boldsymbol{r-2}} \text{ , and} \tag{18}$$

$$\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{vr}} = \boldsymbol{\delta_2} * \boldsymbol{\sigma}(\boldsymbol{W}^{\boldsymbol{M}} \widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{r-1}}) \otimes \boldsymbol{M}_{\boldsymbol{t}}^{\boldsymbol{vr-2}} \text{ .} \tag{19}$$

Where $\boldsymbol{\delta_1, \delta_2} \in \boldsymbol{R}$ and rounds $\boldsymbol{r} = \{\boldsymbol{1}, \ldots, \boldsymbol{R}\}$ are hyper-parameters. In the previous research, a fixed value of $\mathbf{2}$ was used for $\boldsymbol{\delta_1, \delta_2}$. It enables the transformed data after the sigmoid activation function to close to identity [20]. However, the setting is not appropriate due to the difference of KT datasets. Therefore, we optimize $\boldsymbol{\delta_1, \delta_2}$ for each dataset to ensure the best performance. $\boldsymbol{r}$ represents the number of multiplication rounds of $\widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{r}}$ and $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{vr}}$. When $\boldsymbol{r} = \mathbf{1}$, the $\widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{-1}} = \boldsymbol{v_t}$ and $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{v0}} = \widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{v}}$ and the hypernetwork won't be utilized [20]**.** By conducting multiple rounds of multiplications as shown in equations (18) and (19), the hypernetwork optimizes the balance between $\widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{r}}$ and $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{vr}}$. The rounds $\boldsymbol{r}$ is optimized for each dataset. Details of hyper-parameters are presented in the section of Experiment.

## 3.2 Modified memory updating component

By increasing the relevant information, the hypernetwork better optimized the balance between $\widetilde{\boldsymbol{v}}_{\boldsymbol{t}}^{\boldsymbol{r}}$ and $\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}^{\boldsymbol{vr}}$. Continuing with the previous structure leads to the loss of the optimized information obtained from the hypernetwork. Meanwhile, as the parameters $\mathbf{a_t}$ and $\mathbf{e_t}$, which update the value matrix $\boldsymbol{M}_{\boldsymbol{t}}^{\boldsymbol{v}}$, are solely calculated based on the current response, learners with the same response will receive the similar degree of updating. As a result, the model can not accurately distinguish

learners with varying knowledge levels. To address the issues, we propose a modified memory updating component. The updating parameters $\mathbf{a}_t$, $\mathbf{e_t}$ and value matrix at the next time point $\boldsymbol{M_{t+1}^v}$ are calculated as shown below:

$$\mathbf{e_t} = \boldsymbol{\sigma}(\boldsymbol{W^{e1}\tilde{v}_t^r + W^{e2}\tilde{M}_t^{vr} + b^e}) \text{ ,} \tag{20}$$

$$\boldsymbol{z_t} = \boldsymbol{\sigma}(\boldsymbol{W^{z1}\tilde{v}_t^r + W^{z2}\tilde{M}_t^{vr} + b^z}) \text{ ,} \tag{21}$$

$$\mathbf{a_t} = \boldsymbol{tanh}(\boldsymbol{W^{a1}}\, \boldsymbol{z_t} + \boldsymbol{W^{a2}\tilde{M}_t^{vr} + b^a}) \text{ , and} \tag{22}$$

$$\boldsymbol{M_{t+1}^v} = \boldsymbol{\tilde{M}_t^{vr}} \otimes (\mathbf{1} - \boldsymbol{w_t e_t}) + \boldsymbol{w_t}\mathbf{a_t} \text{ .} \tag{23}$$

By incorporating the new input $\boldsymbol{\tilde{M}_t^{vr}}$, the degree of updating for $\boldsymbol{M_t^v}$ will be optimized by considering both the current and past responses [34]. The learners' knowledge statuses, stored in $\boldsymbol{M_t^v}$, will be updated accurately by introducing their previous knowledge levels. As a result, the ability parameters of learners with varying knowledge levels will be better distinguished. Furthermore, we add a new layer $\mathbf{z_t}$ for helping optimize layer $\mathbf{a_t}$ [34].

The proposed method follows a typical deep learning method. Trainable parameters are updated using the back-propagation algorithm based on the loss [34]. The loss is calculated by the binary cross-entropy, which is a commonly used metric for binary classification tasks. The function is shown below:

$$\boldsymbol{loss} = -\sum_t (\boldsymbol{y_t log P_{tj}} + (\mathbf{1} - \boldsymbol{y_t}) \boldsymbol{log}(\mathbf{1} - \boldsymbol{P_{tj}})) . \tag{24}$$

Where $\boldsymbol{y_t} \in \mathbf{(0,1)}$ donates the true value at time point $\boldsymbol{t}$ and **0,1** indicates correct or incorrect [34].

## 3.3 Adversarial training

Furthermore, we propose the incorporation of AT in the model. AT was initially introduced in the field of CV and has been widely adopted in various deep learning tasks [22,23,24]. The function is shown below:

$$\boldsymbol{\eta} = \boldsymbol{\epsilon}\frac{\nabla_x J(\boldsymbol{\rho},\boldsymbol{x},\boldsymbol{y})}{\|\nabla_x J(\boldsymbol{\rho},\boldsymbol{x},\boldsymbol{y})\|_2} \text{ , and} \tag{25}$$

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta} \text{ .} \tag{26}$$

Where $\boldsymbol{x}$ donates the embedded vector of input data and $\boldsymbol{y}$ donates the target labels. $\boldsymbol{\rho}$ donates the parameters of model. $\boldsymbol{J}$ donates the loss of neural networks. $\boldsymbol{\nabla}$ donates the vector differential operator. $\boldsymbol{\eta}$ donates the adversarial sample which is made from the gradient of input data. $\|\cdot\|_2$ donates the L2 norm of the matrix. $\boldsymbol{\epsilon}$ donates the hyper-parameter which controls the sample's impact to $\boldsymbol{x}$. $\widetilde{\boldsymbol{x}}$ donates the new input data of the model.

As the proposed method incorporates more parameters to optimize the memory updating process, the parameters complexity increases. Moreover, the small scale of dataset in KT results in significant differences in data distribution between training and test sets. These factors contribute to the over-fitting problem in the proposed method.

Although various approaches have been proposed to alleviate over-fitting [37,38,39], AT was reported to have the best performance among them [22,23,24,36]. L1 and L2 regularization are the most representative methods. They introduce regularization terms into the loss function [40]. The regularization terms constrain the complexity of the model parameters to improve the robustness [40]. The dropout addresses the over-fitting issue by randomly deactivating certain neurons during training [40]. These approaches essentially alleviate over-fitting by reducing parameter complexity, which prevents the model from being fully trained on the training set [40]. However, these approaches result in the loss of valuable features at the same time [40].

Nonetheless, AT alleviates the over-fitting problem by introducing perturbations to input data. The perturbations alter the input data distribution, enabling model to extract more generalized features from the dataset [22,23,24]. AT forces the parameters not to over-fit the training data [24]. As a result, the model will be robust to the noisy data and the data in different distributions. AT does not result in the loss of valuable features and it is contrary to the previous methods that decrease the complexity of parameters [24]. Furthermore, since perturbations are generated from the gradient, the new input data become harder for the model to predict [23,24]. As a result, AT also improves the model's predictive power when faced with more difficult data [23,24].

To validate the effectiveness of AT, we conduct a simple experiment on the ASSISTments2009 dataset, which has a small scale and where the model exhibited over-fitting. The results are presented in the Table 1. Although previous approaches are observed to slow down the training set fitting process during training, they did not exhibit superior performance on the test set. In contrast, AT remarkably improves the accuracy of the proposed method on the test set. The improvement can be attributed to the distinctive approach of AT in alleviating over-fitting and its ability to improve the prediction accuracy for the difficult data.

TABLE 1
Prediction Accuracy of Alleviating Over-fitting Approaches

| Dataset | metrics | Proposed method | dropout | L1-regularization | L2-regularization | AT |
|---|---|---|---|---|---|---|
| ASSISTments2009 ( with both Item and Skill Inputs) | AUC | 82.55 | 82.52 | 82.58 | 82.42 | **82.82** |
| | Acc | 77.42 | 77.31 | 77.38 | 77.44 | **77.48** |

# Chapter 4: Experiments

In this section, we conducted experiments to compare the prediction accuracy of the proposed method with the mainstream deep learning methods: DKVMN, Deep-IRT, AKT and Deep-IRT (***Tsutsumi et al. 2021***). Moreover, we performed analytical experiments to support the claim that the proposed method also improve the interpretability.

The 5-fold cross validation is applied on the datasets. Each fold comprises 20% of the data for test sets, 20% for validation sets, and 60% for training sets. The proposed method trains the model using training sets and tune the hyper-parameters using the validation sets, and results on test sets will be presented. We compare the performance on six benchmark datasets: ASSISTments2009, ASSISTments2015, ASSISTments2017, Statics2011, Junyi, and Eedi. The detailed information is presented in Table 2 and below:

1. ASSISTments datasets (including ASSISTments2009, ASSISTments2015, ASSISTments2017) are collected from online tutoring systems and have been widely used as benchmark datasets for KT.

2. Statics2011 is collected from an engineering statics course.

3. Eedi is collected from a Korean mathematics education platform which contains the data from 2018 to 2020. The multiple skills combination will be transformed to an unique number.

TABLE 2
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 | 82.8% | 345.0 |
| Eedi | 80000 | 1200 | 27613 | 64.3% | 177.0 |

4. Junyi is collected from the Chinese online learning system called Junyi Academy.

## 4.1 Hyper-parameter selection

We conducted experiments to optimize hyper-parameters mentioned before for each dataset. We performed experiments to determine the optimal number of rounds ***r*** for each dataset and results are presented in Table 3.

TABLE 3
Prediction Accuracy and Hyper-Parameters ***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 | 81.20 | 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** | 78.84 | 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 | **77.69** | 76.74 |
| Eedi (item & skill) | **79.27** | N/A | N/A | N/A | N/A | N/A |

1. $\{\boldsymbol{\delta_1}, \boldsymbol{\delta_2}\}$ are tuned as {**1.5,1.5**} for ASSISTments2009, ASSISTments2015 and ASSISTments2017, {**1.0, 1.7**} for Statics2011, {**1.0, 1.0**} for Junyi and Eedi.

2. As shown in Table 3, AUC score reaches its highest level when **r = 2** for Statics2011, ASSISTments2017 and Junyi with only skill inputs, **r = 3** for ASSISTments2009 and ASSISTments2017 with only skill inputs, **r = 6** for ASSISTments2009 and ASSISTments2017 with both item and skill inputs. As Eedi datasets has a numerous size, the model suffers from the problem of gradient exploding for **r >= 3**.

3. $\boldsymbol{\lambda}$ determines the number of previous value matrices $\{\boldsymbol{M_t^v}, \boldsymbol{M_{t-1}^v}, \ldots, \boldsymbol{M_{t-\lambda}^v}\}$ that should be included as input. We conducted experiments to optimize $\boldsymbol{\lambda}$ based on the optimal setting of $\{\boldsymbol{\delta_1}, \boldsymbol{\delta_2}\}$ and **r**. Results show that $\boldsymbol{\lambda}$ is optimized as $\boldsymbol{\lambda} = 1$ for ASSISTments2009 and Junyi with only skill inputs, and as ASSISTments2009 and ASSISTments2017 with both item and skill inputs. For the remaining datasets, $\boldsymbol{\lambda} = 0$. The hyper-parameters of the previous methods are set according to the optimal settings in the corresponding papers.

TABLE 4
Prediction Accuracy of Student's Performance with Only Skill Inputs

| Datasets | metrics | DKVMN | Deep-IRT | AKT | Deep-IRT (*Tsutsumi et al. 2021*) | Proposed method | Proposed method with AT | ϵ |
|---|---|---|---|---|---|---|---|---|
| ASSISTments2009 | AUC | 81.21 +/- 0.31 | 81.34 +/- 0.39 | 80.81 +/- 0.41 | 81.34 +/- 0.24 | **81.83 +/- 0.30** | **81.92 +/- 0.36** | 1.0 |
| | Acc | 75.11 +/- 0.66 | 76.55 +/- 0.45 | 76.57 +/- 0.55 | **76.91 +/- 0.24** | 76.80 +/- 0.49 | **76.95 +/- 0.19** | |
| ASSISTments2015 | AUC | 72.61 +/- 0.16 | 72.53 +/- 0.23 | **72.97 +/- 0.12** | 72.34 +/- 0.13 | 72.95 +/- 0.14 | **73.06 +/- 0.21** | 0.5 |
| | Acc | 75.05 +/- 0.18 | 74.97 +/- 0.14 | **75.25 +/- 0.10** | 74.95 +/- 0.39 | 75.02 +/- 0.15 | 75.03 +/- 0.14 | |
| ASSISTments2017 | AUC | 72.67+/- 0.37 | 72.08 +/- 0.32 | 73.25+/- 0.41 | 72.32+/- 0.69 | **85.06 +/- 1.17** | **85.63 +/- 1.08** | 1.0 |
| | Acc | 68.46 +/- 0.24 | 68.36 +/- 0.30 | 69.17+/- 0.70 | 68.07 +/- 0.54 | **79.11 +/- 1.06** | **79.26+/- 0.95** | |
| Statics2011 | AUC | 81.20 +/- 0.42 | 81.38 +/- 0.42 | 82.15 +/- 0.35 | 81.45 +/- 0.45 | **82.25 +/- 0.55** | **82.47 +/- 0.47** | 0.2 |
| | Acc | 79.24 +/- 0.84 | 80.33 +/- 0.78 | 80.41 +/- 0.67 | 79.18 +/- 0.67 | **80.63 +/- 0.85** | **80.69 +/- 0.79** | |
| Junyi | AUC | 78.59 +/- 0.21 | 78.39 +/- 0.20 | 78.84 +/- 0.19 | 78.47 +/- 0.21 | **79.00 +/- 0.26** | **79.50 +/- 0.24** | 1.0 |
| | Acc | 86.61 +/- 0.28 | 86.57 +/- 0.30 | 86.54 +/- 0.25 | 86.58 +/- 0.27 | **86.76 +/- 0.24** | **86.82 +/- 0.26** | |
| Eedi | AUC | 75.11 +/- 0.16 | 75.63 +/- 0.17 | **75.81 +/- 0.15** | 75.76 +/- 0.17 | 75.53 +/- 0.15 | 75.60 +/- 0.21 | 1.0 |
| | Acc | 71.23 +/- 0.24 | 71.34 +/- 0.29 | 71.38 +/- 0.20 | **71.41 +/- 0.25** | 71.30 +/- 0.24 | 71.36 +/- 0.25 | |
| Average | AUC | 76.89 | 76.83 | 77.30 | 76.91 | **79.35** | **79.70** | |
| | Acc | 74.46 | 75.05 | 76.55 | 76.18 | **78.27** | **78.36** | |

## 4.2 Results

1. With only skill inputs: The results of Accuracy and AUC for benchmark datasets are presented in Table 4. The results demonstrate the proposed method outperforms previous methods in terms of average AUC and Accuracy. Moreover, the proposed method exhibits significant improvements on datasets with longer response sequences, such as ASSISTments2017 and Statics2011. It aligns with the previous research and supports the notion that the hypernetwork plays a crucial role in the improvement [20].

2. With both item and skill inputs: Furthermore, we compared the performance of the proposed method with AKT and Deep-IRT (***Tsutsumi et al. 2021***) for ASSISTments2009, ASSISTments2017, and Eedi datasets with both item and skill inputs. The results are presented in Table 5. The proposed method consistently exhibits the highest accuracy on average. AKT shows the highest accuracy on Eedi, which can be attributed to the effectiveness of self-attention mechanisms for larger datasets, whereas RNN-based models tend to be relatively weaker with larger datasets. To validate the effectiveness of the modified memory updating component, we conduct experiments on the proposed method without it. The result shows a significant decrease in the prediction accuracy. In dataset with longer response

TABLE 5
Prediction Accuracy of Student's Performance with both Item and Skill Inputs

| Datasets | metrics | AKT | Deep-IRT (Tsutsumi et al. 2021) | Proposed method without modified component | Proposed method | Proposed method with AT | ϵ |
|---|---|---|---|---|---|---|---|
| ASSISTments2009 | AUC | 82.20 +/- 0.25 | 80.70 +/- 0.56 | 82.29 +/- 0.28 | **82.55 +/- 0.32** | **82.82 +/- 0.39** | 1.0 |
| | Acc | 77.30 +/- 0.55 | 76.13 +/- 0.58 | 77.34+/- 0.53 | **77.42 +/- 0.49** | **77.48 +/- 0.49** | |
| ASSISTments2017 | AUC | 74.54+/- 0.21 | 74.15+/- 0.27 | 76.52 +/- 0.36 | **77.69 +/- 0.51** | **77.94 +/- 0.59** | 0.2 |
| | Acc | 69.83+/- 0.15 | 68.73+/- 0.11 | 71.07 +/- 0.24 | **72.16 +/- 0.55** | **72.39 +/- 0.63** | |
| Eedi | AUC | **79.42 +/- 0.11** | 79.11 +/- 0.14 | 79.16 +/- 0.13 | 79.27 +/- 0.15 | 79.34 +/- 0.17 | 1.0 |
| | Acc | **73.59 +/- 0.16** | 73.42 +/- 0.24 | 73.40 +/- 0.19 | 73.49 +/- 0.27 | 73.52 +/- 0.31 | |
| Average | AUC | 78.72 | 78.00 | 79.60 | **79.83** | **80.03** | |
| | Acc | 73.57 | 72.76 | 73.96 | **74.36** | **74.46** | |

TABLE 6
Correlation Coefficient's of The Estimated Abilities

| | No. items | 50 | 100 | 200 | 300 | 50 | 100 | 200 | 300 | 50 | 100 | 200 | 300 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| σ | 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 |
| | Deep-IRT (Tsutsumi) | 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 method | **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 | 0.654 | 0.676 | 0.692 |
| | Deep-IRT (Tsutsumi) | 0.827 | 0.891 | 0.883 | 0.890 | 0.863 | 0.926 | 0.941 | 0.945 | 0.671 | 0.755 | 0.758 | 0.761 |
| | proposed method | **0.840** | **0.905** | **0.900** | **0.907** | **0.877** | **0.932** | **0.947** | **0.954** | **0.720** | **0.755** | **0.768** | **0.779** |
| 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 |
| | Deep-IRT (Tsutsumi) | 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 method | **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 |
| | Deep-IRT (Tsutsumi) | 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 method | **0.854** | **0.840** | **0.854** | **0.836** | **0.894** | **0.920** | **0.930** | **0.919** | **0.708** | **0.744** | **0.762** | **0.743** |

sequences, it even shows a greater decrease. The result validates that the modified memory updating component helps the hypernetwork to utilize the optimized data more efficiently.

We also conduct experiments on datasets to demonstrate the effectiveness of AT. The experimental setup and results are presented in Table 4 and 5. AT consistently improves the performance of the proposed method across all datasets. It indicates the effectiveness of AT in alleviating over-fitting and improving model's robustness. However, the improvement on large-scale datasets such as Eedi is not as significant as on smaller datasets. It could be attributed to the abundance of data available in Eedi for training the model.

## 4.3 Analysis of interpretable parameters

1. Estimation of ability parameters

In this section, we conduct experiments on the simulation dataset to evaluate the performance of interpretability in learners' abilities. Similar to Deep-IRT (***Tsutsumi et al. 2021***) [10,35], the simulation data is generated from the TIRT model, which is a time-series IRT [32,33,34]. The probability of a correct answer assigned to itme $\boldsymbol{j}$ by student $\boldsymbol{i}$ at time $\boldsymbol{t}$ with ability parameter $\boldsymbol{\theta_{it}}$ is assumed as

$$\boldsymbol{P_{ij}(x_{ij} = 1|\theta_{it}) = \frac{1}{1+exp(-\tilde{a}_{\Delta_t}\theta_{it}-b_j)}} \textbf{ ,and} \quad (27)$$

$$\boldsymbol{\tilde{a}_{\Delta_t} = \frac{1}{\sqrt{1+\sigma a_j^2 \Delta_t}}} \quad (28)$$

Where $\boldsymbol{\Delta_t = t - t_j}$ and $\boldsymbol{\tilde{a}_{\Delta_t} \in (0, \infty)}$ is the discrimination parameter at time $\boldsymbol{t}$. $\boldsymbol{b_j \in (-\infty, \infty)}$ is the $\boldsymbol{j}$-th item's difficulty parameter representing the degree of difficulty [31,32]. $\boldsymbol{\theta_{it} \in (-\infty, \infty)}$ represents the ability of student $\boldsymbol{i}$ at time $\boldsymbol{t}$. The prior of $\boldsymbol{\theta_{it}}$ is a normal distribution described as $\boldsymbol{\theta_{i0} \sim N(0, 1)}$ and $\boldsymbol{\theta_{it} \sim N(\theta_{it-1}, \epsilon)}$ [31,32]. $\boldsymbol{\sigma}$ is a variance of $\boldsymbol{\theta_{it}}$ and a forgetting parameter [31,32].

For the experiments, we used datasets consisting of 2000 learners' responses to {**50, 100, 200, 300**} items. The datasets are divided into a 90% training set and a 10% test set [34]. We compared the estimated parameters of the test set. The hyper-parameter $\boldsymbol{\sigma}$ = {**0.1, 0.3, 0.5, 1.0**} controls the degree of learners' forgetting [34]. A larger value of $\boldsymbol{\sigma}$ indicates a greater potential for changes in learners' abilities over time [34]. The Pearson's correlation coefficients, the Spearman's rank correlation coefficients, and the Kendall's rank correlation coefficients which calculated between the ability generated by TIRT and estimated parameters of models. In this experiment, we employed the following three metrics [34]. Each model calculates Spearman's rank correlation is the non-parametric version of the Pearson's correlation. The Kendall's rank correlation provides robust estimates for the aberrant value. Because the distribution of learners' abilities varies over time,

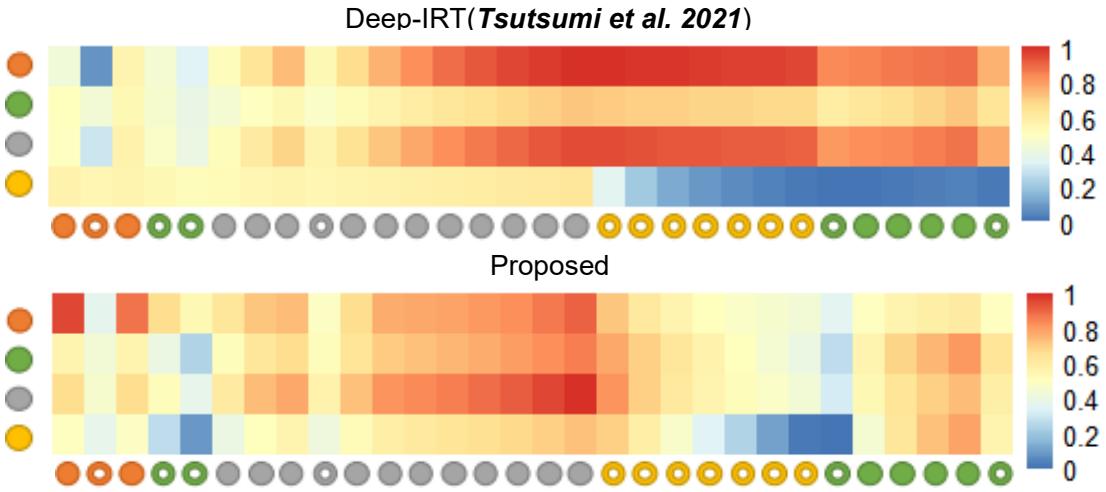


Figure 4: Example of a learner's ability transition from the ASSISTments2009 dataset. The comparison between Deep-IRT(***Tsutsumi et al. 2021***) and the proposed method.The filled and the hollow circles respectively represent correct and incorrect responses.

the root mean square error (RMSE) is not employed as a metric in this case [34].

The correlation coefficients are calculated based on learners' abilities at time $t \in \{1, 2, ..., T\}$, and the average of coefficients is presented in Table 6. The proposed method outperforms to previous methods in all the datasets. The results demonstrate the proposed method improves the performance in accurately capturing learners' abilities and better distinguishing the abilities of learners with varying knowledge levels.

2. Analysis of learner ability transitions

In this section, we evaluate estimated learners' ability transitions of the proposed method and compare them with Deep-IRT (***Tsutsumi et al. 2021***) [10,35]. Visualized ability transition graphs assist teachers to analyze learners' knowledge levels in each skill. Similar to previous researches, we estimate a learner's ability which from the ASSISTments2009 dataset and provide our evaluation of the results [7,8,31,32].

In Figure 4, we present the comparison of Deep-IRT (***Tsutsumi et al. 2021***) and the proposed method. The vertical axis shows the learner's ability value and the range is shown on the right side. The horizontal axis shows four different skills. The

skill inputs are labeled 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.

In the overall view, the proposed methods improve the model's ability to capture learners ability transitions and accurately present the correlations between multi-dimensional skills. Additionally, there are distinct changes observed in the ability of each individual skill. In contrast, the abilities in Deep-IRT (***Tsutsumi et al. 2021***) are relatively independent. The observation indicates that the proposed method effectively captures correlations between different skills and provides more traceable estimation of learner's ability.

In the microscopic view, there is a strong connection between equation solving with more than two steps (grey) and equation solving with two or fewer steps (green). The results demonstrate the proposed method establish a similar pattern between them. Moreover, the proposed method exhibits a stronger influence of the current response on the learner's ability. As the balance between past and latest responses are optimized by the proposed method, learners' abilities receive a stronger impact from the current response and the effect from premature data is also reduced.

# Chapter 5: Conclusions

Knowledge Tracing (KT) is the task of assessing learners' proficiency in skills based on their study history. It has attracted significant attention due to its ability for its capacity to improve learners' learning effectiveness in the field of education.

This research proposed to incorporate a hypernetwork and a modified memory updating component. The modified memory updating component with a hypernetwork improved Deep-IRT model's capability to balance the current input data and past latent variables. With the proposed method, the memory updating component obtained more relevant data for updates and could adaptively update for learners with varying knowledge levels. Additionally, we introduced adversarial training to improve the model's robustness against over-fitting and achieve higher accuracy. Experimental results on benchmark datasets demonstrated that the proposed method outperformed existing methods. Consistent with prior studies [20], the proposed method exhibited significant efficacy, especially for datasets with longer sequences. Additionally, we conducted experiments on simulation data generated from the TIRT model. The results demonstrated that the ability parameters estimated by the proposed method exhibited higher correlations in all situations. The results also indicated that the proposed method improves both the interpretability and performance of the model.

However, the proposed method with a higher complexity structure encounters the challenge of gradient exploding. The model might experience breakdown in the later stages of training. Simplifying the hypernetwork to reduce the model's complexity without compromising accuracy becomes an important task for the future research.

# Acknowledgement

I am grateful to complete my Master's degree at the University of Electro-Communications. Over the three years in lab, Prof. Ueno not only taught me the knowledge related to research, but also instilled in me the essence of research and a rigorous attitude towards everything. It allows me to develop and grow in various aspects of my abilities, empowering me to face future challenges independently. I would like to express my sincere appreciation to Prof. Ueno for his guidance and support.

I would also like to express my appreciation to Prof. Uto. With the issues pointed out by Prof. Uto, I was able to better improve my experimental content and research. Moreover, even with the busy schedule, Prof. Uto always helps me resolve the problems I encountered. I sincerely express my gratitude for his assistance.

I would like to express my appreciation to Mr. Sekiguchi and Ms. Tsutsumi. Their assistance has been invaluable to me throughout my research and daily life. I could not have completed my research during my Master's without their help, and I am truly grateful for their support.

Furthermore, I would like to express my appreciation to the Chinese Students Association of UEC. They have played a significant role in helping Chinses students, including me, adapt to the unfamiliar environment abroad. The activities they organized have greatly enriched my experience, and I hope that the Chinese Students Association will continue to thrive and assist more Chinese students in the future.

I am deeply appreciative of my parents' unwavering support. Despite not being physically present, they have always provided me with moral and material assistance to pursue my studies. I am thankful for their nurturing and hope to be able to give back to them more in the future.

Finally I would like to express my appreciation to all the staff of UEC. They have always extended a helping hand whenever I needed it, ensuring that I never felt helpless in a foreign country. Their support has been instrumental, and I am truly thankful for their kindness and assistance.

# References

[1] CORBETT, Albert T.; ANDERSON, John R. Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 1994, 4: 253-278.

[2] PARDOS, Zachary A.; HEFFERNAN, Neil T. Modeling individualization in a bayesian networks implementation of knowledge tracing. In: User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings 18. Springer Berlin Heidelberg, 2010. p. 255-266.

[3] Zachary A. Pardos and Neil 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), pp. 243–254, 01 2011.

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

[5] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized bayesian knowledge tracing models. In Artificial Intelligence in Education, pp. 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pp. 505–513. Curran Associates, Inc., 2015.

[7] YEUNG, Chun-Kit. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. arXiv preprint arXiv:1904.11738, 2019.

[8] ZHANG, Jiani, et al. Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web. 2017. p. 765-774.

[9] GHOSH, Aritra; HEFFERNAN, Neil; LAN, Andrew S. Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020. p. 2330-2339.

[10] TSUTSUMI, Emiko; KINOSHITA, Ryo; UENO, Maomi. Deep-IRT with Independent Student and Item Networks. International Educational Data Mining Society, 2021.

[11] BARTOLUCCI, Francesco; PENNONI, Fulvia; VITTADINI, Giorgio. Assessment of school performance through a multilevel latent Markov Rasch model. Journal of Educational and Behavioral Statistics, 2011, 36.4: 491-522.

[12] MOLENAAR, Dylan, et al. Hidden Markov item response theory models for responses and response times. Multivariate behavioral research, 2016, 51.5: 606-626.

[13] PARK, Jong Hee. Modeling preference changes via a hidden Markov item response theory model. Handbook of Markov Chain Monte Carlo, 2011, 479-491.

[14] Xiaojing Wang, James Berger, and Donald Burdick. Bayesian analysis of dynamic item response models in educational testing. The Annals of Applied Statistics, Vol. 7, No. 1, pp. 126–153, 2013.

[15] 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.

[16] PANDEY, Shalini; KARYPIS, George. A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837, 2019.

[17] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems, 2017, 30.

[18] H. David, D. Andrew, and V. L. Quoc, “Hypernetworks,” arXiv preprint arXiv:1609.09106, 2016.

[19] STANLEY, Kenneth O.; D’AMBROSIO, David B.; GAUCI, Jason. A hypercube-based encoding for evolving large-scale neural networks. Artificial life, 2009, 15.2: 185-212.

[20] MELIS, Gábor; KOČISKÝ, Tomáš; BLUNSOM, Phil. Mogrifier lstm. arXiv preprint arXiv:1909.01792, 2019.

[21] KRAUSE, Ben, et al. Multiplicative LSTM for sequence modelling. arXiv preprint arXiv:1609.07959, 2016.

[22] GOODFELLOW, Ian J.; SHLENS, Jonathon; SZEGEDY, Christian. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[23] LOWD, Daniel; MEEK, Christopher. Adversarial learning. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. 2005. p. 641-647.

[24] GUO, Xiaopeng, et al. Enhancing knowledge tracing via adversarial training. In: Proceedings of the 29th ACM International Conference on Multimedia. 2021. p. 367-375.

[25] EMBRETSON, Susan E.; REISE, Steven P. Item response theory. Psychology Press, 2013.

[26] Kevin H. Wilson, Yan Karklin, Bojian Han, and Chaitanya 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, pp. 539–544, 06 2016.

[27] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. Neural computation, 1997, 9.8: 1735-1780.

[28] AI, Fangzhe, et al. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. International Educational Data Mining Society, 2019.

[29] Su, Yu, et al. "Exercise-enhanced sequential modeling for student performance prediction." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

[30] LIU, Qi, et al. Ekt: Exercise-aware knowledge tracing for student performance prediction. IEEE Transactions on Knowledge and Data Engineering, 2019, 33.1: 100-115.

[31] SANTORO, Adam, et al. Meta-learning with memory-augmented neural networks. In: International conference on machine learning. PMLR, 2016. p. 1842-1850.

[32] Chaitanya Ekanadham and Yan Karklin. T-skirt: Online estimation of student proficiency in an adaptive learning system. CoRR, Vol. abs/1702.04282, 2017.

[33] Kevin H. Wilson, Yan Karklin, Bojian Han, and Chaitanya 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, pp. 539–544, 06 2016.

[34] Emiko Tsutsumi, Yiming Guo, and Maomi 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.

[35] Emiko Tsutsumi, Ryo Kinoshita, and Maomi Ueno. Deep item response theory as a novel test theory based on deep learning. Electronics, Vol. 10, No. 9, 2021.

[36] WU, Dongxian; XIA, Shu-Tao; WANG, Yisen. Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems, 2020, 33: 2958-2969.

[37] SRIVASTAVA, Nitish, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 2014, 15.1: 1929-1958.

[38] KUKAČKA, Jan; GOLKOV, Vladimir; CREMERS, Daniel. Regularization for deep learning: A taxonomy. arXiv preprint arXiv:1710.10686, 2017.

[39] RICE, Leslie; WONG, Eric; KOLTER, Zico. Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning. PMLR, 2020. p. 8093-8104.

[40] YING, Xue. An overview of overfitting and its solutions. In: Journal of physics: Conference series. IOP Publishing, 2019. p. 022022.