Incremental Cognitive Diagnosis for Intelligent Education

Shiwei Tong
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
tongsw@mail.ustc.edu.cn

Jiayu Liu
School of Data Science, University of Science and Technology of China
Hefei, China
jy251198@mail.ustc.edu.cn

Yuting Hong
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
yutingh@mail.ustc.edu.cn

Zhenya Huang
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
huangzhy@ustc.edu.cn

Le Wu
Hefei University of Technology & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
lewu.ustc@gmail.com

Qi Liu*
School of Computer Science and Technology, University of Science and Technology of China & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
qiliuql@ustc.edu.cn

Wei Huang
School of Data Science, University of Science and Technology of China
Hefei, China
ustc0411@mail.ustc.edu.cn

Enhong Chen
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
cheneh@ustc.edu.cn

Dan Zhang
iFLYTEK Research, iFLYTEK CO., LTD.
Hefei, China
danzhang@iflytek.com

ABSTRACT

Cognitive diagnosis, aiming at providing an approach to reveal the proficiency level of learners on knowledge concepts, plays an important role in intelligent education area and has recently received more and more attention. Although a number of works have been proposed in recent years, most of contemporary works acquire the traits parameters of learners and items in a transductive way, which are only suitable for stationary data. However, in the real scenario, the data is collected online, where learners, test items and interactions usually grow continuously, which can rarely meet the stationary condition. To this end, we propose a novel framework, Incremental Cognitive Diagnosis (ICD), to tailor cognitive diagnosis into the online scenario of intelligent education. Specifically, we first design a Deep Trait Network (DTN), which acquires the trait parameters in an inductive way rather than a transductive way. Then, we propose an Incremental Update Algorithm (IUA) to balance the effectiveness and training efficiency. We carry out Turning Point (TP) analysis to reduce update frequency, where we derive the minimum update condition based on the monotonicity theory of cognitive diagnosis. Meanwhile, we use a momentum update strategy on the incremental data to decrease update time without sacrificing effectiveness. Moreover, to keep the trait parameters as stable as possible, we refine the loss function in the incremental updating stage. Last but no least, our ICD is a general framework which can be applied to most of contemporary cognitive diagnosis models. To the best of our knowledge, this is the first attempt to investigate the incremental cognitive diagnosis problem with theoretical results about the update condition and a tailored incremental learning strategy. Extensive experiments demonstrate the effectiveness and robustness of our method.

CCS CONCEPTS

• Information systems → Data stream mining; **• Social and professional topics → Computing education**.

KEYWORDS

Cognitive Diagnosis; Incremental Learning; Transductive Learning; Inductive Learning

ACM Reference Format:
Shiwei Tong, Jiayu Liu, Yuting Hong, Zhenya Huang, Le Wu, Qi Liu, Wei Huang, Enhong Chen, and Dan Zhang. 2022. Incremental Cognitive Diagnosis for Intelligent Education. In *Proceedings of the 28th ACM SIGKDD*

[*] Corresponding Author

KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539399

Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539399

1 INTRODUCTION

Recently, intelligent education systems have been widely used, where cognitive diagnosis is one of the key fundamental technologies supporting these systems [13, 14]. Cognitive diagnosis can be used to profile learners by discovering their latent cognitive proficiency on knowledge concepts and can also be applied to reveal some traits of the test items such as *difficulty* and *discrimination* [31]. As shown in the left-top part of Figure 1, learners usually first answer a set of test items and leave their responses (e.g., right or wrong) which form up the response matrix, and then a Cognitive Diagnosis Model (CDM) is used to diagnose the trait features of learners and items. In past decades, many CDMs like Item Response Theory (IRT) [17], Deterministic Inputs, Noisy "And" gate model (DINA) [4] and NeurlCD [34] are proposed. These methods diagnose the traits of learners/items in a transductive way and therefore are only suitable for the stationary data, where learners, items and interactions are not expected to change.

However, in the real scenario, learners usually answer testing items online in intelligent education systems. The responses are sequentially recorded and formatted into the streaming log data, which can rarely meet the stationary condition that the transductive CDMs can be applied. As illustrated in the left part of Figure 1, when the incremental logs come, there are more interactions, which might lead to increment of the number of learners and items (i.e., the green blocks). Thus, in order to update the traits of learners/items, transductive CDMs should refit all data including the incremental ones leading to extremely low efficiency (i.e., *branch II* in Figure 1). Although in other areas, some modifications such as Incremental Learning have been proposed to tailor the transductive methods into online scenarios (e.g., incremental matrix factorization), these methods cannot be applied in cognitive diagnosis. Some theories of cognitive diagnosis require the methods to guarantee some psychometric relationships such as monotonicity [24, 31], which are not considered in the popular incremental learning methods. Thus, in the real online scenario of intelligent education, where learners, test items and interactions usually grow continuously, how to apply cognitive diagnosis remains unsolved.

Along this line, there are three challenges to be tackled. First is the new learner/item problem. As mentioned above, with incremental logs arriving, the number of learners and items might increase. Nevertheless, traditional CDMs conducting in a transductive way should refit all data to get the traits of new learners/items (i.e., *area* ④ in Figure 1), which result in low efficiency. Thus, how to find a method to directly deduce the traits of new learners/items becomes a urgent challenge. Second, updating model parameters is a trade-off between effectiveness and efficiency. If we include more data to retrain the CDM (e.g., *branch II* in Figure 1), the prediction effectiveness can be promoted. However, including more data also requires more training time. On the contrary, if we only update on the incremental data (i.e., *branch I* in Figure 1)), the training time can be saved. However, only focusing on a few amount of data leads to potential overfitting, which could result in lower effectiveness. Third, updating model parameters will result in traits unstableness. As shown in the right part of Figure 1, we acquire the original traits (i.e., *area* ①) from accumulated logs and get the updated traits after (i.e., *area* ③) incremental logs arriving. However, as shown in *area* ② of the right part of Figure 1, we find that the updated traits might differ from the original ones of a certain learner/item (highlighted by the red dotted box in *area* ②). Because cognitive diagnosis usually serves as the upstream task, such changes might influence the downstream tasks and analysis [3, 12, 27]. Thus, we hope to keep the trait parameters as stable as possible.

To this end, we propose an Incremental Cognitive Diagnosis (ICD) framework. Specifically, we first design a Deep Trait Network (DTN), where the traits parameters are no longer got in a transductive optimization way but directly deduced from the logs. In this way, we can easily get the traits of new learners/items without refitting on the incremental data. Then, to balance effectiveness and efficiency, we put forward an Incremental Update Algorithm (IUA). We analyze the Turning Point (TP) and derive the minimum update condition based on the monotonicity theory of cognitive diagnosis to reduce update frequency. Meanwhile, we use a momentum update strategy on the incremental data to decrease update time without sacrificing effectiveness. Moreover, to keep the trait parameters as stable as possible, we refine the loss function in the incremental update stage. To the best of our knowledge, this is the first attempt to investigate the incremental cognitive diagnosis problem with theoretical results about the update condition and a tailored incremental learning strategy. Extensive experiments demonstrate the effectiveness and robustness of ICD.

2 RELATED WORK

Cognitive Diagnosis. Cognitive diagnosis is a fundamental but important task in many real-world scenarios such as games[2], medical diagnosis [37], and especially, education [15, 31]. The main goal of cognitive diagnosis is to learn the latent trait features of learners from their testing logs. These learned trait features could be applied to many tasks, such as performance prediction [34] and resource recommendation [3]. In early years, cognitive diagnosis was mostly developed from psychometric [13]. IRT [16] and DINA [4] are the two most fundamental but classic cognitive diagnosis models which model the response result of a learner answering an item as the interaction between the trait features of the learner and the item. By extending the trait features into multidimensional, Reckase et al. [22] proposed Multidimensional Item Response Theory (MIRT). Recently, some researchers introduce the deep learning into cognitive diagnosis [32, 34, 36]. Wang et al. [34] proposed NeuralCD exploiting neural networks to automatically learn the interaction function. Tsutsumi et al. [32] built a learner network and an item network to implement better representation of trait features. Generally, these methods learn the trait parameters in a transductive way, therefore are only suitable for the stationary data. Thus, how to tailor CDMs into the online scenario of intelligent education, where the log data is sequentially collected remains unsolved.

Incremental Learning. Incremental learning investigates how to learn with streaming data in an interactive scenario where training examples are provided over time [7]. It has been developed for several machine learning tasks, such as image classification [20],

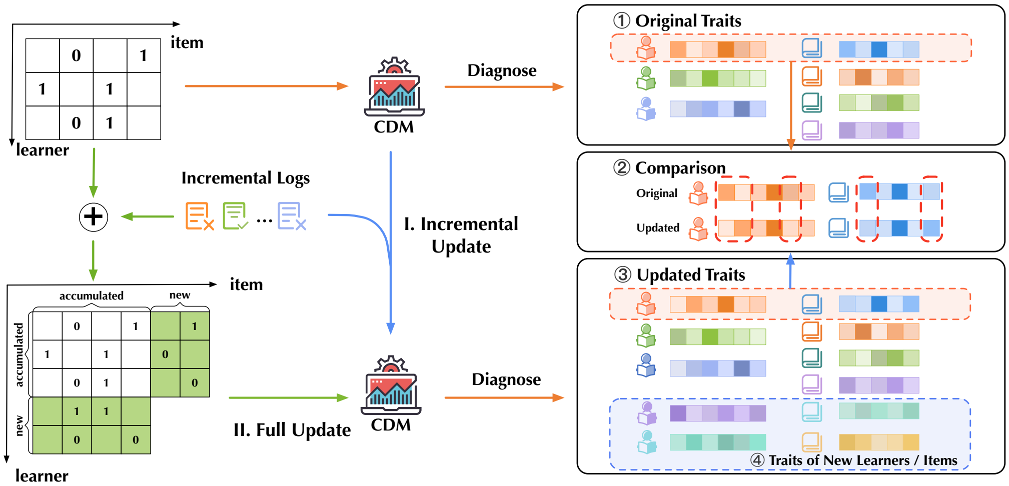


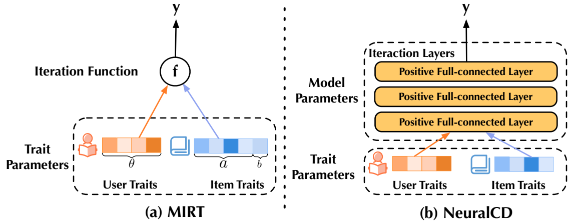
Figure 1: The flowchart of incremental update and full update for CDMs. Branch I. "Incremental Update" means the CDM adjust the trait parameters only using the incremental logs while Branch II. "Full Update" represents the CDM use all data to update the trait parameters.

recommender system [26], and reinforcement learning [1]. For incremental cognitive diagnosis, the problem is formalized similarly to Incremental Matrix Factorization (IMF), as both of them aim at processing a changing matrix and predicting its unknown elements. Following Huang et al. [10], current IMF models can be divided into three categories: SVD-based model, vector-retaining model and space-retraining model. SVD-based models [25] are characterized by using singular value decomposition (SVD) and require the matrix with no missing elements. Vector-retraining models focus on updating [33] or retraining [5, 6, 18, 21, 23, 28] the latent feature vector when incremental data arrives by additional pass of algorithm like stochastic gradient descent (SGD). Space-retraining methods [10, 26, 35] propose to evolve the whole feature matrix in order to reduce redundant calculation. Though these works have achieved great success, there exist some limitations when applied to incremental cognitive diagnosis. First, there is a psychometric relationship (e.g., monotonicity) between matrix elements in incremental cognitive diagnosis (i.e., the traits of learners/items). Second, because cognitive diagnosis usually serves as the upstream task, the trait parameters should be kept as stable as possible. To the best of our knowledge, our paper is the first work towards the incremental cognitive diagnosis problem, which gives theoretical results about the update condition and a tailored incremental learning strategy.

3 PRELIMINARY

Cognitive Diagnosis Models. Before we step into our method, we would like to first briefly introduce Cognitive Diagnosis Models (CDMs). CDMs are developed to depict learner's proficiency level on specific knowledge concepts based on her responses to several test items, i.e., $\boldsymbol{u}, \boldsymbol{v} \longleftarrow R$, where $\boldsymbol{u}, \boldsymbol{v}$ are the latent traits of learners and items while R is the responses data. Usually, the Learner Performance Prediction task [30, 34] is used to learn the trait parameters with the optimization target of minimizing the difference between the predicted probability $P(y_{ij})$ and the true response r_{ij}. Mostly, the cross entropy is used as the loss function:

$$\mathcal{L} = -\sum\nolimits_{i,j}(r_{ij}\,log\,y_{ij} + (1 - r_{ij}\,log(1 - y_{ij}))). \tag{1}$$



(a) MIRT

(b) NeuralCD

Figure 2: Comparison of MIRT and NeurlCD.

In the past decades, some CDMs have been proposed such as DINA and IRT. Generally, CDMs contain two parts: (1) the representations of trait features and (2) the interaction function (also called Item Response Function, IRF). For example, IRT uses single-dimension variables to represent the trait features and logistic function as the interaction function as follows:

$$P(y_{ij}|\theta_i, a_j, b_j) = \frac{1}{1 + e^{-1.7a_j(\theta_i - b_j)}}, \tag{2}$$

where a_j and b_j represent the discrimination and difficulty of item j, and θ_i indicates the proficiency level of the learner i. Using multidimensional vectors to represent latent traits of both test items and learners, IRT is extended to MIRT:

$$P(y_{ij}|\boldsymbol{\theta}_i, \boldsymbol{a}_j, b_j) = \frac{1}{1 + e^{-\boldsymbol{a}_j\boldsymbol{\theta}_i + b_j}}. \tag{3}$$

Recently, some researchers introduce the deep learning into cognitive diagnosis [32, 34, 36]. For example, Wang et al. [34] proposed NeuralCD, which exploits neural networks to automatically learn the interaction function and could be seen as the generalization of many traditional psychometric CDMs. In these deep learning models, extra model parameters are included. As shown in Figure 2, we have the following general form of CDMs:

$$y_{ij} = \mathcal{M}_{CD}(\boldsymbol{u}_i, \boldsymbol{v}_j; \Theta_{CD}), \tag{4}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ respectively represents the trait parameters of learners and items, and Θ_{CD} is the model parameters. To be noticed that, in these models, a psychometric relationship, called monotonicity [24, 31], should be strictly maintained. The monotonicity theory declares that learner's proficiency is monotonic with the

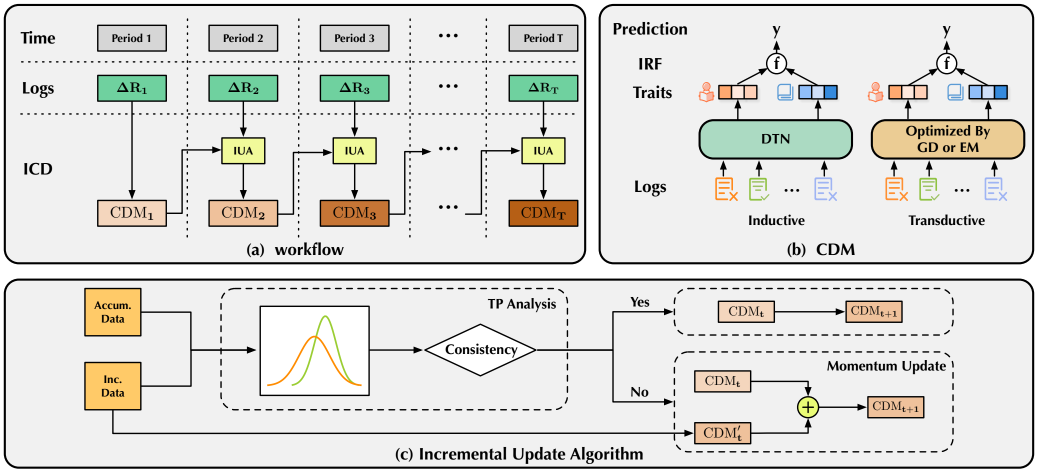


Figure 3: The illustration of ICD framework. The left-top part (a) shows how ICD incrementally learn from the streaming logs, where details of CDM are presented in (b) compared with traditional methods and IUA is illustrated in (c).

probability of giving the right response to a test item, i.e., $\frac{\partial f}{\partial u} > 0$. In addition, some CDMs like DINA and NeuralCD can explicitly use Q-matrix [29] labeled by experts to obtain knowledge-aware latent features, i.e., a certain dimension of the trait can reflect the proficiency on the corresponding concept (e.g., u_{ik} in NeuralCD is the proficiency level of learner i on concept k.).

With Gradient Descent (GD) or Expectation Maximization (EM) algorithm, we can learn the trait parameters and model parameters:

$$\boldsymbol{u}^*, \boldsymbol{v}^*, \Theta_{CD}^* \leftarrow argmin_{\boldsymbol{u},\boldsymbol{v},\Theta_{CD}} \mathcal{L}(R, \mathcal{M}_{CD}(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\Theta_{CD}})). \quad (5)$$

It is easy to know that contemporary CDMs learn the trait parameters in a transductive way.

Problem Definition. In each time period $t+1$, denoting the accumulated logs as $R_t = \Delta R_1 + ... + \Delta R_t$, the learners in R_t as U_t and the items in R_t as V_t. The incremental logs are represented as ΔR_{t+1}, which might contain new learners and new items. We denote the set of new learners as ΔU_t and new items as ΔV_t. Each record in logs is a tuple (u_i, v_j, r_{ij}), where r_{ij} is the score (transferred to binary, i.e., 0 indicates wrong answer while 1, otherwise). Furthermore, we have Q-matrix $Q = \{Q_{jk}\}_{M \times L}$, where $Q_{jk} = 1$ if the item v_j relates to the knowledge concept c_j and $Q_{jk} = 0$ otherwise. M is the total number of items and L is the number of knowledge concepts. Here, we assume the number of learners N and the number of items M may increase with the incremental logs arriving, but the number of knowledge concepts is set to be static. Our goal is to precisely diagnose the trait parameters of all learners (i.e., $U_{t+1} = U_t + \Delta U_t$) and items (i.e., $V_{t+1} = V_t + \Delta V_t$):

Incremental Cognitive Diagnosis Problem. When the incremental logs ΔR_{t+1} comes in the time period $t+1$, our goal is to efficiently and effectively mine learners' proficiency on concepts.

4 INCREMENTAL COGNITIVE DIAGNOSIS

To tailor cognitive diagnosis into the online scenario, we propose an Incremental Cognitive Diagnosis (ICD) framework. In ICD, we aim to solve two key issues: (1) how to accelerate the procedure obtaining the trait parameters of learners or items, especially for new learners/items; (2) how to speed up the incremental update without sacrificing effectiveness. As shown in Figure 3, our ICD includes two main parts: CDM with Deep Trait Networks (DTNs) and Incremental Update Algorithm (IUA). As illustrated in Figure 3(a), we used the streaming logs to train the CDM with DTNs (i.e., the left part of Figure 3(b)), where model parameters of DTNs are used to deduce the trait parameters. Then, as illustrated in Figure 3(c), IUA is used to update the model when incremental logs arrive, which will have two branches based on the information consistency.

4.1 Deep Trait Network

As we discussed above, contemporary CDMs learn the trait parameters in a transductive way which are inefficient to process online streaming data. Thus, we design Deep Trait Networks (DTNs) to acquire the trait parameters in an inductive way. For better illustration, we highlight the difference between our proposed inductive CDM with DTN and traditional transductive CDM in Figure 3(b). Concretely, as shown in Figure 4, our DTNs consist of two independent networks, one is DTN of Learner, i.e., L-DTN, and the other one is DTN of Item, i.e., I-DTN. These two networks share the exactly same architecture, but have different model parameters. Each DTN is composed by three parts: (1) a deep trait embedding layer, (2) a deep trait feature layer and (3) a non-sequential pooling layer. As the two independent networks use the same architecture, we use L-DTN as the example to explain how DTN works in detail.

We assume the input to L-DTN is the intact logs of a learner $u \in U$, which is a disordered interaction tuple sequence $X = \{x_0, x_1, ..., x_n\}$. n is the total number of logs and the interaction tuple $x_k = \{v_k, r_k\}$. In Deep Trait Embedding Layer, we first use a one-hot vector $\boldsymbol{x}_k \in \{0,1\}^{2N}$ to represent the interaction tuple:

$$\boldsymbol{x}_k^j = \begin{cases} 1 & \text{if } j = 2 \cdot v_k + r_k, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

After that, an embeddding layer is used to map the sparse one-hot $\boldsymbol{x}_k$ into a low-dimension dense space, which can save the complexity of model parameters:

$$\mathcal{X}_k = \boldsymbol{x}_k \mathrm{E}_d. \quad (7)$$

$E_d \in \mathbb{R}^{2M \times d_e}$ is a weight matrix and d_e is the embedding dimension.

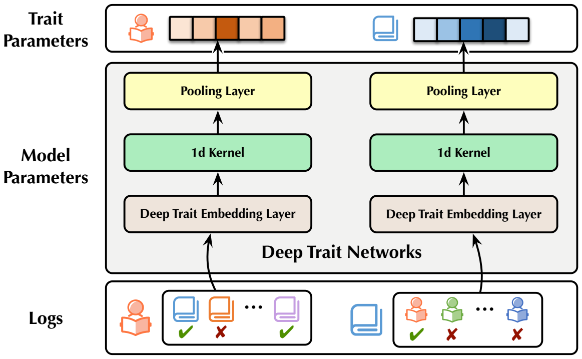


Figure 4: Overview of DTNs.

Then, we use a feature module and a pooling layer to aggregate the whole information of the logs to get the trait parameters. The feature module consists of several 1d convolutional kernels. Each kernel contains a filter $\boldsymbol{w} \in \mathbb{R}^{d_e}$, which is applied to produce a higher-level trait features:

$$\mathcal{F}_j = \sigma(\boldsymbol{w} \cdot \mathcal{X}_j + b). \tag{8}$$

Here $b \in R$ is a bias term and σ is a non-linear activation function such as the relu.

As the numbers of logs differ in different learners, we hope to find a length-free aggregate function to summarize the information contained in the interaction sequence. Note that, the order of different interaction tuples should have no effect on deducing the trait parameters. To this end, we use *average pooling*, an order-free operation, to implement the information aggregation:

$$\mathcal{T}^j = \frac{\sum_{k=0}^{n} \mathcal{F}_k^j}{n}. \tag{9}$$

Obviously, average operation can receive interaction sequence with a arbitrary length as the input. In practice, to reduce the space complexity, we could incrementally calculate $\mathcal{T}_{t+1}^j = \frac{n_t \mathcal{T}_t^j + \sum_{k=0}^{n_{t+1}} \mathcal{F}_k^j}{n_t + n_{t+1}}$ by caching $\mathcal{T}_t^j$ and n_t. As for I-DTN used to acquire the trait parameters of the item, the first element v_k in the interaction tuple x_k should be replaced as the u_k and the other parts are exactly same as L-DTN. With the two DTNs, we can get the trait parameters of the learner u and item v:

$$\begin{aligned} \boldsymbol{u} &= L - DTN(X_u), \\ \boldsymbol{v} &= I - DTN(X_v). \end{aligned} \tag{10}$$

Then we substitute $\boldsymbol{u}$ and $\boldsymbol{v}$ into Eq.(4) and train DTNs with Eq.(1). Once we get well trained DTNs, we can use them for new learner/item trait deducing. For example, for a new learner with logs $X_{u'}$, we can easily apply L-DTN to get her trait parameters in an inductive way, i.e., $\boldsymbol{u'} = L - DTN(X_{u'})$, which is time-efficient.

4.2 Incremental Update Algorithm

In this part, we want to solve the second issue, i.e., how to speed up the incremental updating without sacrificing effectiveness. We put forward an Incremental Update Algorithm (IUA) to address the issue from two aspects: (1) reduce the update frequency and (2) decrease update time.

4.2.1 Turning Point Analysis. Some incremental data has no extra information that can help model to promote the effectiveness. The information brought by these data is well contained in the accumulated data. Therefore, as shown in the middle part of Figure 3(c), in order to reduce the update frequency, we propose to find the Turning Point (TP), which indicates the condition that we should update the model parameters according to the incremental data. To find TP is to figure out when it's necessary to update an item's trait v and a learner's trait u according to the incremental data ΔR. Notice that, when it's necessary to update either an item's or a learner's trait vector, the TP of L-DTN or I-DTN is reached and we need to incrementally learn model parameters.

As the DTNs share the same architecture, here we use I-DTN as the example to carry out the mathematical analysis. Note by symmetry, the analysis can also be applied to find the TP of L-DTN. Before detailed derivation, we first give the former definitions of "Consistent learner" and "Consistent item's record".

Definition 1. ***Consistent learner*** *$i \in \Delta R$: $\forall$ function g, item $j \in R$, $E[g(u_i, v_j)] = \frac{1}{m_j} \sum_{i' \in R} g(u_{i'}, v_j)$.*

Definition 2. ***Consistent item's record*** *$r_{ij} \in \Delta R$: learner i is consistent, $\forall$ function g, $E[g(u_i, v_j, r_{ij})] = \frac{1}{m_j} \sum_{i' \in R} g(u_{i'}, v_j, r_{i'j})$.*

m_j is the number of learners that have answered j in R. Here, "consistency" means that the information in incremental data ΔR has been contained in accumulated data R, and that's why we can imply new learners' features from accumulated records. In DEFINITION 1, a consistent learner i's interaction with each item j is the average of learners who have answered j in R, thus her profile can be seen as sampled from the distribution of existing learners' traits. DEFINITION 2 further considers whether her performance on item j is consistent with existing records. For example, if all the learners in R have answered j correctly, but the consistent learner i answered wrong, then we may say r_{ij} is not consistent with item j's history. We emphasize that analysis should be based on the assumption of "Consistent learner". On the contrary, it's unsuitable to assume "Consistent item's record", because under such condition, current model has performed well on incremental data, and there is no need to update items' profiles any more.

To evaluate the degree of inconsistency of new records, we focus on the losses before and after incremental data coming. Specifically, the training losses related to item j on R and $R \bigcup \Delta R$ are:

$$\begin{aligned} \text{Loss}_R = &-\frac{1}{m_j} \sum\nolimits_{i \in R} r_{ij} \cdot \log f(u_i, v_j) \\ &+ (1 - r_{ij}) \cdot \log(1 - f(u_i, v_j)), \\ \text{Loss}_{R \cup \Delta R} = &-\frac{1}{m_j + n_j} \sum\nolimits_{i \in R \cup \Delta R} r_{ij} \cdot \log f(u_i, v_j) \\ &+ (1 - r_{ij}) \cdot \log(1 - f(u_i, v_j)), \end{aligned} \tag{11}$$

where n_j is the number of learners that answered j in ΔR. On one hand, when the assumption of "Consistent item's record" is true, $E[\text{Loss}_{R \cup \Delta R}] = \text{Loss}_R$, where the randomness comes from the uncertainty of learners' traits $u_i, i \in \Delta R$. On the other hand, if $E[\text{Loss}_{R \cup \Delta R}] < \text{Loss}_R$, current model has generated well on incremental data, and we no longer need to modify model parameters. Therefore, it's reasonable to estimate how much $E[\text{Loss}_{R \cup \Delta R}]$ is larger than Loss_R, and the problem is summarized as below:

Given item j's trait v_j trained well on R, and new responses ΔR, in which the learners are consistent. Ask in what range the updated value Δv of v_j belongs to can we ensure that $E[\mathrm{Loss}_{R\cup\Delta R}(\Delta v)] - \mathrm{Loss}_R < \epsilon$. In particular, we have the following theorem.

Theorem 1. *A necessary condition of v_j for the above problem is*

$$\|\Delta v\| > \frac{\mathrm{KL}\left(p_{\Delta R}\|p_{Avg}\left(f\right)\right) + \mathrm{H}\left(p_{\Delta R}\right) - Loss_R - \frac{m_j+n_j}{n_j}\epsilon}{\left[p\left(|\Delta R^+|\right)\cdot p\left(|R^-|\right) + p\left(|\Delta R^-|\right)\cdot p\left(|R^+|\right)\right]\cdot\frac{\beta_j}{\delta_j}}. \tag{12}$$

We refer the reader to APPENDIX 7.1.1 for the detailed proof, and we present the sketch here. $\mathrm{Loss}_{R\cup\Delta R}(\Delta v)$ is defined as:

$$\begin{aligned}\mathrm{Loss}_{R\cup\Delta R}(\Delta v) = -\frac{1}{m_j+n_j}\sum\nolimits_{i\in R\cup\Delta R} r_{ij}\cdot\log f\left(u_i, v_j+\Delta v\right)\\ +\left(1-r_{ij}\right)\cdot\log\left(1-f\left(u_i, v_j+\Delta v\right)\right).\end{aligned} \tag{13}$$

By the first-order Taylor expansion, it's approximately equal to

$$\begin{aligned}\mathrm{Loss}_{R\cup\Delta R}(\Delta v) \approx -\frac{1}{m_j+n_j}\{-m_j\cdot\mathrm{Loss}_R - n_j\cdot\mathrm{Loss}_{\Delta R}\\ +\sum_{i\in\Delta R}\left[\frac{r_{ij}}{f\left(u_i,v_j\right)} - \frac{\left(1-r_{ij}\right)}{1-f\left(u_i,v_j\right)}\right]\left(\frac{\partial f}{\partial v}\right)^T\Delta v\}.\end{aligned} \tag{14}$$

According to the assumption of "Consistent learner", we can derive a lower bound of $E[\mathrm{Loss}_{\Delta R}]$ as

$$E[\mathrm{Loss}(\Delta R)] \geq \mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right) + \mathrm{H}\left(p_{\Delta R}\right), \tag{15}$$

where $\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right)$ is the KL divergence between distribution of responses on ΔR and predicted results on R, $\mathrm{H}\left(p_{\Delta R}\right)$ is the entropy of $p_{\Delta R}$. Besides, denote the lower bound of $f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)$ as δ_j, the upper bound of $\left\|\frac{\partial f}{\partial v}\right\|$ as β_j, we also have:

$$\begin{aligned}&E\left[\sum_{i\in\Delta R}\frac{r_{ij}-f\left(u_i,v_j\right)}{f\left(u_i,v_j\right)\cdot\left(1-f\left(u_i,v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\Delta v\right]\\ &\leq \frac{n_j\beta_j}{\delta_j}\cdot\left[p\left(\left|\Delta R^+\right|\right)\cdot p\left(|R^-|\right) + p\left(|\Delta R^-|\right)\cdot p\left(\left|R^+\right|\right)\right]\cdot\|\Delta v\|.\end{aligned} \tag{16}$$

In order to ensure $E[\mathrm{Loss}_{R\cup\Delta R}(\Delta v)] - \mathrm{Loss}_R < \epsilon$, combining Eqs. (14),(15) and (16), we finally get

$$\|\Delta v\| > \frac{\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right) + \mathrm{H}\left(p_{\Delta R}\right) - \mathrm{Loss}_R - \frac{m_j+n_j}{n_j}\epsilon}{\left[p\left(|\Delta R^+|\right)\cdot p\left(|R^-|\right) + p\left(|\Delta R^-|\right)\cdot p\left(|R^+|\right)\right]\cdot\frac{\beta_j}{\delta_j}}. \tag{17}$$

Several conclusions can be found from the theorem. First, when the true response distribution on ΔR is similar to the predicted distribution on R, (i.e., $\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right)$ is small), current v_j well fits ΔR, and the demand to update Δv weakens. Meanwhile, there exists a trade-off between it and $\mathrm{H}\left(p_{\Delta R}\right)$. For example, when the records in ΔR are homogeneous (e.g., all responses are "correct"), $\mathrm{H}\left(p_{\Delta R}\right)$ is small but $\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right)$ can be high. Second, when increase n_j, the right part of Eq. (17) becomes higher, indicating that it will be more convincing to update the item's trait with more incremental data. Last but not least, we reiterate that the above derivation can be symmetrically applied to Δu of learner i's profile, which should be based on the assumption of "Consistent item" and "Consistent learner's record".

Definition 3. ***Consistent item*** *$j \in \Delta R$: $\forall$ function g, learner $i \in R$, $E[g(u_i, v_j)] = \frac{1}{m_i}\sum_{j'\in R} g(u_i, v_{j'})$.*

Definition 4. ***Consistent learner's record*** *$r_{ij} \in \Delta R$: item j is consistent, $\forall$ function g, $E[g(u_i, v_j, r_{ij})] = \frac{1}{m_i}\sum_{j'\in R} g(u_i, v_{j'}, r_{ij'})$.*

Here, m_i represents the number of items that i has answered in R. Different from THEOREM 1, we can derive a tighter bound of $\|\Delta u\|$ based on the monotonicity assumption of learners' proficiency:

Theorem 2. *A necessary condition of u_i for the problem is*

$$\|\Delta u\| > \frac{\mathrm{KL}\left(p_{\Delta R}\|p_{Avg}(f)\right) + \mathrm{H}\left(p_{\Delta R}\right) - Loss_R - \frac{m_i+n_i}{n_i}\epsilon}{\sqrt{\left[p\left(|\Delta R^+|\right)\cdot p\left(|R^-|\right)\right]^2 + \left[p\left(|\Delta R^-|\right)\cdot p\left(|R^+|\right)\right]^2}\cdot\frac{\beta_i}{\delta_i}}. \tag{18}$$

The proof is presented in APPENDIX 7.1.2. It should be noticed that monotonicity assumption applies only to learners' profiles, so Eq. (18) is invalid for items. In summary, by setting a threshold ρ for updated value, the update condition of DTN is

Turning Point: *There exists a learner i satisfying* Eq. (19) *or an item j satisfying* Eq. (20):

$$\frac{\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right) + \mathrm{H}\left(p_{\Delta R}\right) - \mathrm{Loss}_R - \frac{m_i+n_i}{n_i}\epsilon}{\sqrt{\left[p\left(|\Delta R^+|\right)\cdot p\left(|R^-|\right)\right]^2 + \left[p\left(|\Delta R^-|\right)\cdot p\left(|R^+|\right)\right]^2}\cdot\frac{\beta_i}{\delta_i}} \geq \rho, \tag{19}$$

$$\frac{\mathrm{KL}\left(p_{\Delta R}\|p_{\mathrm{Avg}}(f)\right) + \mathrm{H}\left(p_{\Delta R}\right) - \mathrm{Loss}_R - \frac{m_j+n_j}{n_j}\epsilon}{\left[p\left(|\Delta R^+|\right)\cdot p\left(|R^-|\right) + p\left(|\Delta R^-|\right)\cdot p\left(|R^+|\right)\right]\cdot\frac{\beta_j}{\delta_j}} \geq \rho. \tag{20}$$

4.2.2 *Momentum Update.*

After we find the turning point, another question remains to be answered: how can we reduce the training time? An intuitive way is to only perform updating based on the incremental data, i.e.,

$$\Theta^{'} \leftarrow \Delta R(\mathcal{M}(\Theta)). \tag{21}$$

However, as we discussed before, such an approach might lower the effectiveness due to the rapidly changing encoder that reduces the key representations' consistency. Thus, inspired by MoCo [9], we use a momentum update strategy to update the model parameters, which is illustrated in the right-bottom part of Figure 3(c). Specifically, the model parameters of DTN before updating are denoted as $\mathbf{\Theta}$ and the model parameters after incrementally updated on the incremental data are denoted as $\mathbf{\Theta}^{'}$. We update $\mathbf{\Theta}$ by:

$$\mathbf{\Theta} \leftarrow \alpha\mathbf{\Theta} + (1-\alpha)\mathbf{\Theta}^{'}. \tag{22}$$

Here $\alpha \in [0, 1)$ is a momentum coefficient. By the momentum update, we can balance the incremental and accumulated information, therefore avoid the overfitting on the incremental data.

4.2.3 *Stableness Penalty.*

By previous steps, we have already implemented an efficient and effective update method from the perspective of accurate prediction. However, we should emphasize that our final target is not only make CDMs well perform on Learner Performance Prediction task [31, 34], but to acquire a robust representations of traits. As we mentioned before, because cognitive diagnosis usually serves as the upstream task, therefore, the trait representations should be kept as stable as possible. Thus, in addition to keep model parameters evolve smoothly, we also need to keep our predicted trait parameters as stable as possible. To achieve this, we add a stableness penalty into Eq.(1) and get the loss function of ICD:

$$\mathcal{L}_{ICD} = \beta\mathcal{L} + (1-\beta)\sum\nolimits_b(\boldsymbol{u}'_b - \boldsymbol{u}_b)^2 + (\boldsymbol{v}'_b - \boldsymbol{v}_b)^2, \tag{23}$$

where $\beta \in (0, 1]$ is a hyper-parameter. $\boldsymbol{u}'_b$ and $\boldsymbol{v}'$ are the trait parameters of the learner and item after incremental updating while $\boldsymbol{u}_b$ and $\boldsymbol{v}_b$ are the original ones.

Statistics	ASSISTments	MATH
# users	4,163	10,268
# items	17,746	917,495
# knowledge concepts	123	1,488
# response logs	324,572	864,722

Table 1: The statistics of two datasets.

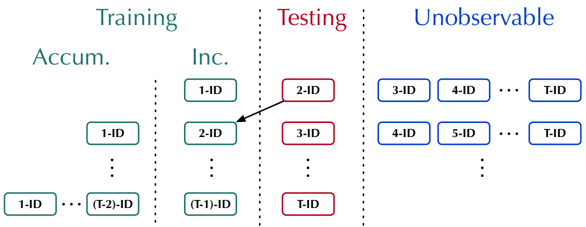


Figure 5: Illustration of experimental process. The incremental training process is shown as the arrow from testing 2-ID to training 2-ID.

5 EXPERIMENTS

5.1 Experimental Setup

We use two real-world datasets in the experiments, i.e., ASSISTments[1] and MATH. ASSISTments (ASSISTments 2009- 2010 "skill builder") is an open dataset collected by the ASSISTments online tutoring systems. The MATH dataset supplied by iFLYTEK Co., Ltd. was collected from Zhixue.com[2] a widely-used online learning system, which contains mathematical test items and logs of high school examinations. Table 1 shows basic statistics of the datasets. We filter out learners with less than 15 and 30 response logs for ASSISTments and MATH respectively to guarantee that each learner has enough data for diagnosis.

Experimental Process. Inspired by Huang et al. [10], as illustrated in Figure 5, the experimental process includes: 1. the experimental data sorted by timestamp is divided into T disjoint continuous parts with a similar scale. We call the i-th incremental data part i-ID; 2. i-ID is used as the incremental training set of CDMs; 3. $(i+1)$-ID is used to simulate the online incremental logs. In addition, $(i+1)$-ID is also used as the testing set to evaluate CDMs; 4. We repeat steps 2 and 3 from $i = 1$ until $i = T - 1$.

Baselines. To evaluate the performance of our ICD, four well-known CDMs, i.e., IRT [17], DINA [4], MIRT [22] and NeuralCD [34], are used as backbone methods. In MIRT, the dimension of latent trait features of both learner and item unitedly is set as 3, while in DINA and NeuralCD, we set the dimension as the number of knowledge concepts, i.e., 123 in ASSISTments and 1488 in MATH. As we discussed in Sec. 2, because advanced incremental learning methods could not directly used for CDMs, therefore, we use two vanilla training strategies. One is "Full Training" (Full) and the other one is "Incremental Training" (Inc.). When incremental data comes, "Full" takes all training sets (i.e., "Accumulated." (Accum.) and "Incremental" (Inc.) in Figure 5) to retrain the model, while "Inc" only takes the incremental data (i.e., "Inc.").

Implementation Details. We initialize parameters in all networks with *Xavier* initialization [8] and we use the Adam algorithm [11] for optimization. In ICD, ϵ is set as 0.01 and ϕ is set as 0.3. In addition,the first $10\% \cdot T$ training sets are used as warmup training set where "Inc" transductive CDMs and ICD-CDMs will be trained in the full training way to implement initialization. All models are implemented by Pytorch using Python and all experiments are run on a Linux server with two Intel(R) Xeon(R) E5-2650 v4 CPUs and a NVIDIA Tesla K80 GPU. Our code is available at https://github.com/bigdata-ustc/EduCDM.

Metrics		ASSISTments			MATH		
		Acc	AUC	DOA	Acc	AUC	DOA
IRT	Inc.	0.652	0.588	-	0.706	0.566	-
	Full	0.713	0.729	-	0.744	0.659	-
	ICD	0.656	0.755	-	0.732	0.727	-
DINA	Inc.	0.517	0.541	0.518	0.454	0.653	0.492
	Full	0.675	0.718	**0.676**	0.563	0.707	0.522
	ICD	0.680	0.676	0.665	0.781	0.759	0.554
MIRT	Inc.	0.645	0.625	-	0.697	0.588	-
	Full	0.714	0.724	-	0.646	0.641	-
	ICD	0.673	0.670	-	0.777	0.774	-
NeuralCD	Inc.	0.657	0.558	0.475	0.770	0.766	0.491
	Full	0.720	0.750	0.643	0.774	0.757	0.516
	ICD	**0.724**	**0.758**	0.664	**0.797**	**0.792**	**0.617**

Table 2: Learner performance prediction results of T-ID.

5.2 Evaluation Metrics

Classification Metrics. Because we cannot obtain the true knowledge proficiency of learners, it is hard to directly evaluate the performance of a cognitive diagnosis model. Following previous works [31, 34], as the diagnostic result is usually acquired through learners performance prediction task, performance on the prediction task can indirectly evaluate the model based on some classification metrics such as *Accuracy*, *AUC*.

Degree of Agreement. Following previous works [31, 34], we adopt Degree of Agreement (DOA) to further investigate the monotonicity based on concepts. Specifically, if learner i has a better mastery on knowledge concept k than learner j, then i is more likely to answer item l related to k correctly than j. For concept k, $DOA(k)$ is formulated as:

$$DOA(k) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\delta(\theta_{ik},\theta_{jk})\sum_{l=1}^{M} I_{lk} \wedge J(l,i,j) \wedge \delta(r_{il},r_{jl})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\delta(\theta_{ik},\theta_{jk})\sum_{l=1}^{M} I_{lk} \wedge J(l,i,j) \wedge [r_{il} \neq r_{jl}]}. \quad (24)$$

θ_{ik} is the proficiency of learner i on concept k. $\delta(x,y) = 1$ if $x > y$ and $\delta(x,y) = 0$ otherwise. $I_{lk} = 1$ if item l contains concept k and $I_{lk} = 0$ otherwise. $J(l,i,j) = 1$ if both learner i and j did item l and $J(l,i,j) = 0$ otherwise. We average $DOA(k)$ on all concepts (i.e., $\overline{DOA}$) to evaluate the quality of diagnostic result.

Stableness An important issue of cognitive diagnosis for the online scenario is to keep the traits as stable as possible after incremental updating. Therefore, we use Manhattan Distance to evaluate models from the perspective of keeping the traits stableness:

$$S = \sum_{i=1}^{N}\sum_{j=1}^{M} |\mathcal{T}_j' - \mathcal{T}_j|/MN, \quad (25)$$

where M is the dimension of traits and N is the instance number of traits, the overall stableness metric is the micro average value of learner traits stableness and item traits stableness, i.e.,

$$\overline{S} = (N_u S_u + N_v S_v)/(N_u + N_v), \quad (26)$$

where N_u and n_v are the instance number of learners and items.

[1] https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data
[2] https://www.zhixue.com/

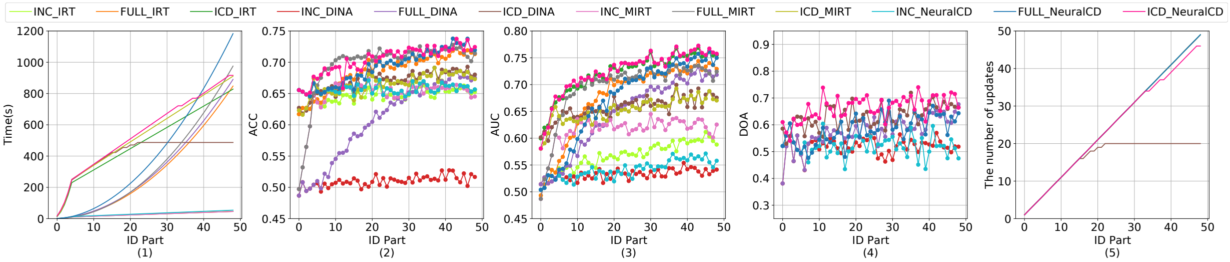


Figure 6: The experimental results of 1-ID to $T-1$-ID on ASSISTments.

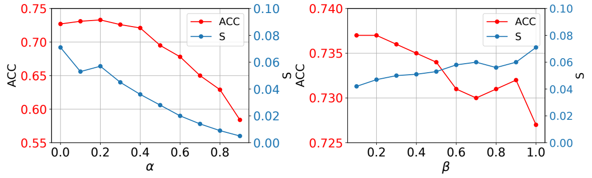


Figure 7: Influence of α and β. α balances the information for accumulated data and incremental data while β balances the prediction accuracy and traits stableness.

5.3 Experimental Results

5.3.1 Learner Performance Prediction. The experimental results of T-id are shown in Table 2. The result of each CDM is recorded in three sub-rows: "Inc.", "Full" and "ICD". For better illustration, we underline the best results for each CDM and bold the best results of all models. From the table, we can find "Full" significantly outperforms "Inc." on all three metrics. This indicates that including more data to retrain the CDM can promote the prediction effectiveness. Meanwhile, we notice that our "ICD" consistently outperforms "Inc." and achieves the best results on almost all metrics, even compared with "Full". These observations indicate that our method can help CDM promote the diagnosis precision (i.e., Acc and AUC) and well maintain the monotonicity (i.e., DOA[3]). Summarily, we have the conclusion that our ICD can help CDMs promote the diagnosis precision with fewer data.

5.3.2 Training Effectiveness and Efficiency. As shown in Figure 6, we can see that CDMs with full training strategy are usually more effective, while the efficiency of them is much lower than the incremental baselines. As for our method, although ICD needs more training time than "Inc." because DTNs contain more model parameters, its performance in terms of effectiveness is quite satisfactory, even though compared with "Full" baselines. Therefore, we could say our ICD realize a good balance between effectiveness and efficiency. Meanwhile, as illustrated in Figure 6(5), we find ICD-CDMs sometimes do not update the model parameters even there is incremental data. The reason why our ICD "skip" some training sets (i.e., fewer update numbers than "Full" and "Inc" CDMs) is that our method has the capacity of detecting whether the information in the incremental data contains in accumulated data by TP analysis. Specially, we find ICD-NeuralCD seems automatically reach an "early stop", which we guess is an another important factor to explain why ICD-CDMs could outperform other baselines. Consequently, we might say the ability of analyzing the turning point can not only reduce update frequency to promote the time efficiency of ICD but also help the model to refine the information contained in the data. In short, ICD can help CDMs achieve the better prediction effectiveness without sacrificing too much training efficiency.

5.3.3 Parameter Sensitive. In ICD, the trade-off parameter α and β play crucial roles in balancing accumulated and incremental information, which greatly influence the model effectiveness and trait stableness. We carry out the parameter sensitive experiments on ASSISTments to see the influence of α and β.

Momentum α. When α is smaller, the model pays more attention the information from the incremental data. Conversely, as α is larger, the model is allowed to focus more on the influence from original data. We perform an experiment on different α, where we set $\beta = 1$ to avoid the influence from β. As shown in the left part of Figure 7, when α increases, the accuracy first increases, but then decreases. This indicates that properly including the information from accumulated data and incremental data is beneficial. When α is too small or too large, the accuracy drops considerably. Meanwhile, we also notice that when α increases, stableness gradually decays, This suggests concentrating more on incremental data will bring in more unstableness. These results prove it is vital to balance the information from accumulated data and incremental data, and support our motivation of using a momentum update strategy.

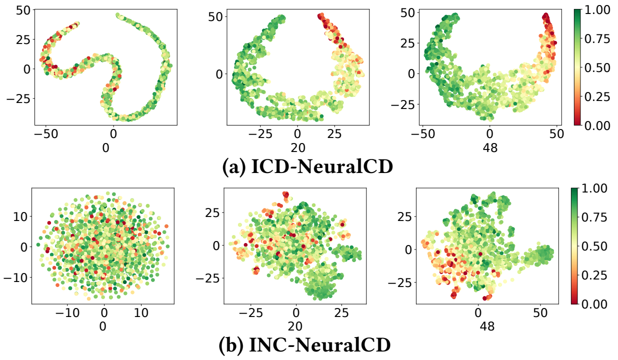
Penalty Weight β. When β is smaller, the model put more stress on maintaining the trait stableness, otherwise, the promoting of prediction accuracy. An experiment on different β is conducted in the condition where $\alpha = 0$ to avoid the influence from α. As shown in right part of Figure 7, when β increases, the accuracy will decrease, while the trait stableness is promoted. Thus, β is a factor balancing the model effectiveness and trait stableness, which can be adjusted based on different needs.

5.3.4 Learner Clustering. We select the learners appearing at the most beginning and visualize the trait representation vectors in different time periods utilizing the T-SNE method [19] to see how the trait representation evolves. We show the trait vectors diagnosed by NeuralCD with two training strategies in Figure 8. From Figure 8(a), we can see, those learners with higher proficiency (i.e. the green ones) are grouped into the left part, while the learners with lower proficiency are grouped to the right. This suggests that the trait representation learned by our method could effectively reflect the learner proficiency. Meanwhile, compared with NeuralCD trained by "Inc" (i.e., Figure 8(b)), the trait vectors obtained by our method evolves more smoothly, which proves that our method can well maintain the stableness of traits.

6 CONCLUSION

In this paper, we proposed a novel Incremental Cognitive Diagnosis (ICD) framework for intelligent education. Specifically, we designed

[3] The reason why IRT and MIRT do not have the results on DOA is that there are no clear correspondence between their latent features and knowledge concepts.



(a) ICD-NeuralCD

(b) INC-NeuralCD

Figure 8: Learner traits clustering in different time periods. The deeper green color indicates the higher proficiency while the deeper red color represents the lower proficiency.

the DTN to acquire the trait parameters of learners and items in an inductive way, which can solve the new learner/item problem. Meanwhile, to balance the predicting effectiveness and training efficiency, we proposed an Incremental Update Algorithm (IUA), where a turning point was mathematically given and a momentum update strategy was introduced. Furthermore, we added the stableness penalty to the loss function to keep the stableness of traits. As a general framework, ICD can be applied to most of CDMs. Extensive experiments demonstrate the effectiveness and robustness of ICD. In the future, we will continue working on exploring how to further reduce the time complexity and space complexity of our method. Meanwhile, we are going to exploit the monotonicity to pretrain DTNs. In addition, we would like to apply our ICD to some other online diagnosis scenarios such as games and medical diagnosis.

Acknowledgements. This research was partially supported by grants from the National Key Research and Development Program of China (No. 2021YFF0901003), and the National Natural Science Foundation of China (Grant No. 61922073, U20A20229 and 62106244). Le Wu greatly thanks the support from the Young Elite Scientists Sponsorship Program by CAST and ISZS.

REFERENCES

[1] A. Barreto, D. Precup, and J. Pineau. On-line reinforcement learning using incremental kernel-based stochastic factorization. *Advances in Neural Information Processing Systems*, 25:1484–1492, 2012.

[2] S. Chen and T. Joachims. Predicting matchups and preferences in context. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 775–784, 2016.

[3] Y. Chen, X. Li, J. Liu, and Z. Ying. Recommendation system for adaptive learning. *Applied psychological measurement*, 42(1):24–41, 2018.

[4] J. De La Torre. Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 34(1):115–130, 2009.

[5] R. Devooght, N. Kourtellis, and A. Mantrach. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 189–198, 2015.

[6] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, and W. Nejdl. Real-time top-n recommendation in social streams. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 59–66, 2012.

[7] A. Gepperth and B. Hammer. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*, 2016.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[9] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[10] X. Huang, L. Wu, E. Chen, H. Zhu, Q. Liu, Y. Wang, and B. T. I. Center. Incremental matrix factorization: A linear feature transformation perspective. In *IJCAI*, pages 1901–1908, 2017.

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

[12] Y.-W. Lee and Y. Sawaki. Cognitive diagnosis approaches to language assessment: An overview. *Language Assessment Quarterly*, 6(3):172–189, 2009.

[13] Q. Liu. Towards a new generation of cognitive diagnosis. In Z. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4961–4964. ijcai.org, 2021.

[14] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang. Exploiting cognitive structure for adaptive learning. In A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 627–635. ACM, 2019.

[15] Q. Liu, R. Wu, E. Chen, G. Xu, Y. Su, Z. Chen, and G. Hu. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(4):1–26, 2018.

[16] F. Lord. A theory of test scores. *Psychometric monographs*, 1952.

[17] F. M. Lord. *Applications of item response theory to practical testing problems.* Routledge, 1980.

[18] X. Luo, Y. Xia, and Q. Zhu. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27:271–280, 2012.

[19] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[20] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[21] P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. Gama. Forgetting methods for incremental matrix factorization in recommender systems. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 947–953, 2015.

[22] M. D. Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pages 79–112. Springer, 2009.

[23] S. Rendle and L. Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 251–258, 2008.

[24] P. R. Rosenbaum. Testing the conditional independence and monotonicity assumptions of item response theory. *Psychometrika*, 49(3):425–435, 1984.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, volume 1, pages 27–8. Citeseer, 2002.

[26] Q. Song, J. Cheng, and H. Lu. Incremental matrix factorization via feature space re-learning for recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 277–280, 2015.

[27] M. A. Sorrel, J. Olea, F. J. Abad, J. de la Torre, D. Aguado, and F. Lievens. Validity and reliability of situational judgement test scores: A new approach based on cognitive diagnosis models. *Organizational Research Methods*, 19(3):506–532, 2016.

[28] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *The Journal of Machine Learning Research*, 10:623–656, 2009.

[29] K. K. Tatsuoka. Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach. *Cognitively diagnostic assessment*, pages 327–359, 1995.

[30] N. Thai-Nghe, T. Horváth, and L. Schmidt-Thieme. Factorization models for forecasting student performance. In *Educational Data Mining 2011*. Citeseer, 2010.

[31] S. Tong, Q. Liu, R. Yu, W. Huang, Z. Huang, Z. Pardos, and W. Jiang. Item response ranking for cognitive diagnosis. IJCAI, 2021.

[32] E. Tsutsumi, R. Kinoshita, and M. Ueno. Deep-irt with independent student and item networks. *International Educational Data Mining Society*, 2021.

[33] J. Vinagre, A. M. Jorge, and J. Gama. Fast incremental matrix factorization for recommendation with positive-only feedback. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 459–470. Springer, 2014.

[34] 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 AAAI Conference on Artificial Intelligence*, volume 34, pages 6153–6161, 2020.

[35] F. Wang, H. Tong, and C. Y. Lin. Towards evolutionary nonnegative matrix factorization. In *25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, AAAI-11/IAAI-11*, pages 501–506, 2011.

[36] M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.

[37] J. Xu, C. Deng, X. Gao, D. Shen, and H. Huang. Predicting alzheimer's disease cognitive assessment via robust low-rank structured sparse model. In *IJCAI: proceedings of the conference*, volume 2017, page 3880. NIH Public Access, 2017.

7 APPENDIX

7.1 Proof of Turning Point

7.1.1 Proof of THEOREM 1. For $\text{Loss}_{R\cup\Delta R}(\Delta v)$, we do a first-order Taylor expansion of $\log f\left(u_i, v_j+\Delta v\right), \log\left(1-f\left(u_i, v_j+\Delta v\right)\right)$:

$$\begin{aligned}\log f\left(u_i, v_j+\Delta v\right) &\approx \log f\left(u_i, v_j\right)+\frac{1}{f\left(u_i, v_j\right)}\left(\frac{\partial f}{\partial v}\right)^T \Delta v, \\ \log\left(1-f\left(u_i, v_j+\Delta v\right)\right) &\approx \log\left(1-f\left(u_i, v_j\right)\right)-\frac{1}{1-f\left(u_i, v_j\right)}\left(\frac{\partial f}{\partial v}\right)^T \Delta v.\end{aligned} \tag{27}$$

As v_j is the locally optimal solution of Loss_R, we also have:

$$\sum_{i\in R}\left[r_{ij}\cdot\frac{1}{f\left(u_i, v_j\right)}-\left(1-r_{ij}\right)\cdot\frac{1}{1-f\left(u_i, v_j\right)}\right]\left(\frac{\partial f}{\partial v}\right)^T=0. \tag{28}$$

Combined with Eqs. (27),(28), $\text{Loss}_{R\cup\Delta R}(\Delta v)$ is approximated by

$$\begin{aligned}\text{Loss}_{R\cup\Delta R}(\Delta v) \approx -\frac{1}{m_j+n_j}\{&-m_j\cdot\text{Loss}_R-n_j\cdot\text{Loss}_{\Delta R} \\ &+\sum_{i\in\Delta R}\left[\frac{r_{ij}}{f\left(u_i, v_j\right)}-\frac{\left(1-r_{ij}\right)}{1-f\left(u_i, v_j\right)}\right]\left(\frac{\partial f}{\partial v}\right)^T\Delta v\}.\end{aligned} \tag{29}$$

To estimate $E[\text{Loss}_{R\cup\Delta R}(\Delta v)]$, we have to evaluate $E[\text{Loss}_{\Delta R}]$, $E[\sum_{i\in\Delta R}\frac{r_{ij}-f(u_i,v_j)}{f(u_i,v_j)\cdot(1-f(u_i v_j))}(\frac{\partial f}{\partial v})^T\Delta v]$ respectively. On one hand, due to the assumption of "Consistent learner", $E[\text{Loss}_{\Delta R}]$ equals to

$$\begin{aligned}&E[-\frac{\sum_{i\in\Delta R} r_{ij}\cdot\log f\left(u_i, v_j\right)+\left(1-r_{ij}\right)\cdot\log\left(1-f\left(u_i, v_j\right)\right)}{n_j}] \\ &=-\frac{1}{n_j}\sum_{i\in\Delta R}\left[\frac{r_{ij}}{m_j}\sum_{i'\in R}\log f\left(u_{i'}, v_j\right)+\frac{\left(1-r_{ij}\right)}{m_j}\sum_{i'\in R}\log\left(1-f\left(u_{i'}, v_j\right)\right)\right].\end{aligned} \tag{30}$$

Define $p\left(\left|\Delta R^+\right|\right)\triangleq\frac{1}{n_j}\sum_{i\in\Delta R} r_{ij}, p\left(\left|\Delta R^-\right|\right)\triangleq\frac{1}{n_j}\sum_{i\in\Delta R}(1-r_{ij})$ are the proportions of correct/wrong answers in ΔR; $\text{Avg}(f_R^+)\triangleq\frac{1}{m_j}\sum_{i'\in R} f(u_{i'}, v_j), \text{Avg}(f_R^-)\triangleq\frac{1}{m_j}\sum_{j'\in R}(1-f(u_{i'}, v_j))$ are the average predicted probabilities of learners in R correctly/wrongly responding j. From the Geometric-Means Inequality,

$$\begin{aligned}E[\text{Loss}(\Delta R)] \geq &-\left[p\left(\left|\Delta R^+\right|\right)\cdot\log(\text{Avg}(f_R^+))+p\left(\left|\Delta R^-\right|\right)\cdot\log\left(\text{Avg}(f_R^-)\right)\right] \\ &=\text{KL}\left(p_{\Delta R}\|p_{\text{Avg}}(f)\right)+\text{H}\left(p_{\Delta R}\right).\end{aligned} \tag{31}$$

On the other hand, $E[\sum_{i\in\Delta R}\frac{r_{ij}-f(u_i,v_j)}{f(u_i,v_j)\cdot(1-f(u_i v_j))}(\frac{\partial f}{\partial v})^T\Delta v]$ can be calculated similarly:

$$\begin{aligned}&E[\sum_{i\in\Delta R}\frac{r_{ij}-f\left(u_i, v_j\right)}{f\left(u_i, v_j\right)\cdot\left(1-f\left(u_i, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\Delta v] \\ &=\left[\sum_{i\in\Delta R}\frac{r_{ij}}{m_j}\sum_{i'\in R}\frac{1}{f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right. \\ &\left.-\frac{n_j}{m_j}\cdot\sum_{i'\in R}\frac{1}{\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right]\cdot\Delta v \\ &=\left[\frac{n_j}{m_j}\cdot\sum_{i'\in R}\frac{p\left(\left|\Delta R^+\right|\right)-f\left(u_{i'}, v_j\right)}{f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right]\cdot\Delta v.\end{aligned} \tag{32}$$

Combined with Eq. (28), Eq. (32) further becomes

$$\begin{aligned}&E[\sum_{i\in\Delta R}\frac{r_{ij}-f\left(u_i, v_j\right)}{f\left(u_i, v_j\right)\cdot\left(1-f\left(u_i, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\Delta v] \\ &=\left[\frac{n_j}{m_j}\cdot\sum_{i'\in R}\frac{p\left(\left|\Delta R^+\right|\right)-r_{i'j}}{f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right]\cdot\Delta v \\ &=\frac{n_j}{m_j}\cdot\left[\sum_{i'\in R^-}\frac{p\left(\left|\Delta R^+\right|\right)}{f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right. \\ &\left.-\sum_{i'\in R^+}\frac{p\left(\left|\Delta R^-\right|\right)}{f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\right]\cdot\Delta v.\end{aligned} \tag{33}$$

Denote the lower bound of $f\left(u_{i'}, v_j\right)\cdot\left(1-f\left(u_{i'}, v_j\right)\right)$ as δ_j, the upper bound of $\left\|\frac{\partial f}{\partial v}\right\|$ as β_j, then Eq. (33) is bounded by:

$$\begin{aligned}&E[\sum_{i\in\Delta R}\frac{r_{ij}-f\left(u_i, v_j\right)}{f\left(u_i, v_j\right)\cdot\left(1-f\left(u_i, v_j\right)\right)}\left(\frac{\partial f}{\partial v}\right)^T\Delta v] \\ &\leq\frac{n_j\beta_j}{\delta_j}\cdot\left[p\left(\left|\Delta R^+\right|\right)\cdot p\left(\left|R^-\right|\right)+p\left(\left|\Delta R^-\right|\right)\cdot p\left(\left|R^+\right|\right)\right]\cdot\|\Delta v\|.\end{aligned} \tag{34}$$

In order to ensure $E[\text{Loss}_{R\cup\Delta R}(\Delta v)]-\text{Loss}_R<\epsilon$, combine Eqs. (29),(31) and (34), we finally get

$$\|\Delta v\|>\frac{\text{KL}\left(p_{\Delta R}\|p_{\text{Avg}}(f)\right)+\text{H}\left(p_{\Delta R}\right)-\text{Loss}_R-\frac{m_j+n_j}{n_j}\epsilon}{\left[p\left(\left|\Delta R^+\right|\right)\cdot p\left(\left|R^-\right|\right)+p\left(\left|\Delta R^-\right|\right)\cdot p\left(\left|R^+\right|\right)\right]\cdot\frac{\beta_j}{\delta_j}}. \tag{35}$$

Note that an example of the ideal situation is that all the responses in R and ΔR are "correct" ("wrong"), and the predicted probabilities of the model trained on R are 1 (0). Then, $\text{KL}\left(p_{\Delta R}\|p_{\text{Avg}}(f)\right)+\text{H}\left(p_{\Delta R}\right)$ equals 0, and Eq. (35) holds naturally. Under such situation, we do not need to update the item's trait v_j.

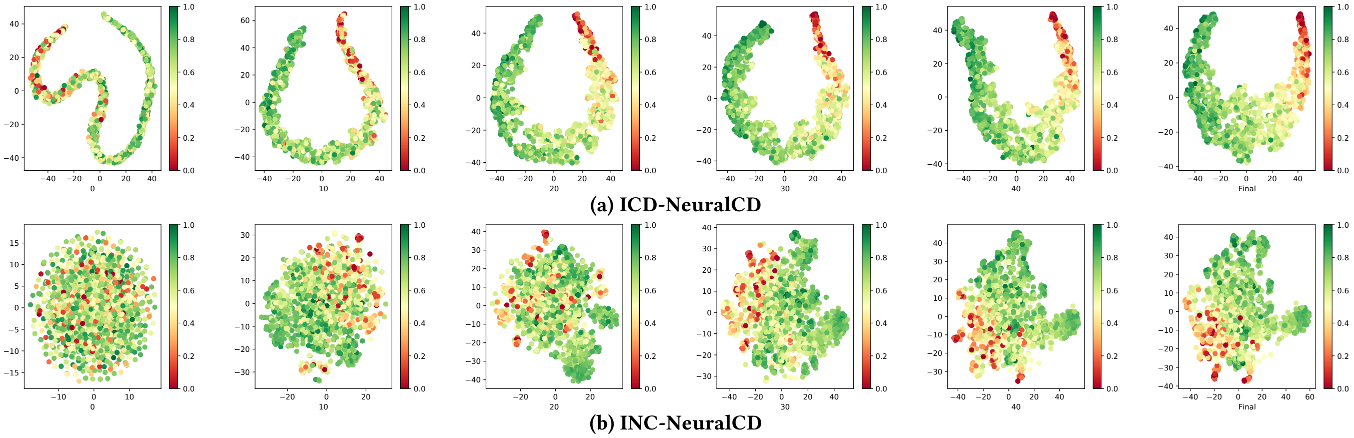
7.1.2 Proof of THEOREM 2. The above analysis can be similarly applied to learner u. However, according to monotonicity assumption, each element of $\frac{\partial f}{\partial u}$ is positive. Thus by denoting $\Delta u^+\triangleq(max\{\Delta u_k, 0\}, k=1,2,...,|\Delta u|), \Delta u^-\triangleq-(min\{\Delta u_k, 0\}, k=1,2,...,|\Delta u|)$ $(\Delta u=\Delta u^+-\Delta u^-)$, we have $-\left\|\frac{\partial f}{\partial u}\right\|\cdot\|\Delta u^-\|\leq(\frac{\partial f}{\partial u})^T\cdot\Delta u=(\frac{\partial f}{\partial u})^T\cdot\Delta u^+-(\frac{\partial f}{\partial u})^T\cdot\Delta u^-\leq\left\|\frac{\partial f}{\partial u}\right\|\cdot\left\|\Delta u^+\right\|$. A tighter upper bound of Eq. (34) is therefore derived as follows:

$$\begin{aligned}&E[\sum_{j\in\Delta R}\frac{r_{ij}-f\left(u_i, v_j\right)}{f\left(u_i, v_j\right)\cdot\left(1-f\left(u_i, v_j\right)\right)}\left(\frac{\partial f}{\partial u}\right)^T\Delta u] \\ &\leq\frac{n_i\beta_i}{\delta_i}\cdot\left[p\left(\left|\Delta R^+\right|\right)\cdot p\left(\left|R^-\right|\right)\cdot\left\|\Delta u^+\right\|+p\left(\left|\Delta R^-\right|\right)\cdot p\left(\left|R^+\right|\right)\cdot\|\Delta u^-\|\right] \\ &\leq\frac{n_i\beta_i}{\delta_i}\cdot\sqrt{\left[p\left(\left|\Delta R^+\right|\right)\cdot p\left(\left|R^-\right|\right)\right]^2+\left[p\left(\left|\Delta R^-\right|\right)\cdot p\left(\left|R^+\right|\right)\right]^2}\cdot\|\Delta u\|.\end{aligned} \tag{36}$$

The last inequality in Eq. (36) comes from Cauthy-Schwarz Inequality and the fact that $\|\Delta u\|^2=\|\Delta u^+\|^2+\|\Delta u^-\|^2$.

7.2 Detailed Result of Learner Clustering.

We show the detailed evolution of Sec. 5.3.4 in Figure 9. It is easy to see that, the trait vectors got by our method get grouped at 10-ID while the vectors got by INC-NeuralCD get grouped until



(a) ICD-NeuralCD

(b) INC-NeuralCD

Figure 9: Learner traits clustering in different time periods. The deeper green color indicates the higher proficiency while the deeper red color represents the lower proficiency.

30-ID. Meanwhile, we could find that the trait vectors got by our method keep the same relevant location along with the incremental training, while the vectors acquired by the baseline changes a lot. Thus, we could say that our method can get the accurate trait vectors more efficiently and make the trait vectors stable along with the incremental training.