Behavioral Testing of Deep Neural Network Knowledge Tracing Models

Minsam Kim
Riiid
Seoul, South Korea
minsam.kim@riiid.co

Yugeun Shim
Riiid
Seoul, South Korea
yugeun.shim@riiid.co

Seewoo Lee
Riiid
Seoul, South Korea
seewoo.lee@riiid.co

Hyunbin Loh
Riiid
Seoul, South Korea
hb.loh@riiid.co

Juneyoung Park
Riiid
Seoul, South Korea
juneyoung.park@riiid.co

ABSTRACT

Knowledge Tracing (KT) is a task to model students' knowledge based on their coursework interactions within an Intelligent Tutoring System (ITS). Recently, Deep Neural Networks (DNN) showed superb performance over classical methods on multiple dataset benchmarks. While most Deep Learning based Knowledge Tracing (DLKT) models are optimized for general objective metrics such as accuracy or AUC on benchmark data, proper deployment of the service requires additional qualities. Moreover, the black-box nature of DNN models makes them particularly difficult to diagnose or improve when unexpected behaviors are encountered. In this context, we adopt the idea of black-box testing / behavioral testing from Software Engineering and (1) define desirable KT model behaviors to (2) propose a KT model analysis framework to diagnose the model's behavioral quality. We test-run the framework using three state-of-the-art DLKT models on seven datasets based on the proposed framework. The result highlights the impact of dataset size and model architecture upon the model's behavioral quality. The assessment results from the proposed framework can be used as an auxiliary measure of the model performance by itself, but can also be utilized in model improvements via data-augmentation, architecture design, and loss formulation.

Keywords

Knowledge Tracing, Deep Learning, Behavioral Testing, Model Validation

1. INTRODUCTION

Assessment is a central task in Education, as it is involved in meta-cognition [17], tracing the skill trajectory, recommendation of contents [36], adjustment of tutoring strategy [14], and grading [3, 24, 33, 35]. With the advent of online educational platforms, there is an increasing demand in building assessment models using the interaction history data of users. One approach to track the skill of users is Knowledge Tracing (KT), which is the task to model students knowledge based on their coursework interactions within an Intelligent Tutoring System (ITS) [7].

To tackle the KT problem, the recent EdNet Challenge in Kaggle has gathered a total of 3,406 teams, 4,412 participants, to submit 64,678 models. Participants trained KT models on the EdNet KT dataset [6], and the models were evaluated by the Area Under the Receiver Operating Characteristic Curve (AUC). The AUC of the top 5 models were 0.820, 0.818, 0.818, 0.817, 0.817, which are very similar, and all models were based on the Transformer Neural Network structure [38]. While neural network structures are usually designed to appropriate human intuitions, most models lack interpretability compared to classical models. Therefore, when evaluation results for black-box models do not vary significantly, it becomes unclear how to choose the best model for deployment. Also, while small quantitative difference in the objective function or model AUC might not hurt the users' perception of the model reliability, few adversarial decisions of the black-box model can dissuade the user's faith. [34] also note that the performances of black-box models that are trained for general metrics such as classification accuracy or AUC(Area Under receiving operator characteristic Curve) can be overestimated.

As a result, Deep Learning based Knowledge Tracing (DLKT) models are not frequently implemented in the education community due to potential risks arising from the lack of model interpretability. In this study, we propose behavioral testing as an approach to alleviate this problem. The contribution of this work are summarized as follows:

- We propose a novel testing framework to validate DLKT models using a test on behaviors. The idea is to define consistent and convincing behaviors to be desired on DLKT models.
- As an example of applying the framework, we benchmark three state-of-the-art DLKT models from the

proposed validation framework. Positive results highlight the reliability of DLKT models and encourages the model's adoption, while negative results point out the limitations of DLKT models and show spaces for improvement.

- We introduce methods to utilize evaluations from the framework to design and improve DLKT models.

2. RELATED WORKS

2.1 Knowledge Tracing

Knowledge Tracing (KT) is the task to predict the expected correctness of an interaction of a student to a question by modeling the student's knowledge from past interactions [7]. In this study, we formulate the KT task as follows: the interaction sequence of a user is denoted as $X^u = \{x_1^u, x_2^u, \cdots, x_T^u\}$ where $u \in U$ is the user index. To simplify the notations, we omit the user index u unless specified. Each interaction $x_t = (q_{i_t}, c_t)$ at step t is defined by the pair of question $q_{i_t} \in Q$ and correctness $c_t \in \{0, 1\}$ where $Q = \{q_1, q_2, \cdots, q_n\}$ denotes the set of all questions and i_t denotes the question index of step t. A KT model predicts the correctness probability $P(c_t = 1|x_1, x_2, \cdots, x_{t-1}, q_{i_t})$ of an unseen interaction X_t at step t, where $X_t = x_1, x_2, \cdots, x_t$ is the first t interactions of an interaction sequence X.

Notation	Description
$u \in U$	User index
X	Interaction sequence of a single user($= X^u$)
x_t	Interaction at time step t
$q_j \in Q$	Question
i_t	Question index at time step t
c_t	Correctness at time step t for question q_{i_t}
$c_t^{(q_j)}$	Correctness at time step t for question q_j
X_t	Interaction sequence of X up to time step t

Many KT models utilize domain specific tags such as skill components of questions [27, 8, 31, 43, 42], difficulty of questions [32, 10, 37], or knowledge graphs [4]. Item Response Theory (IRT) [32] models the correctness probability of a student responding to a question using custom designed models, and fits the model parameters using maximum likelihood. For instance, the 4-PL model predicts the correctness probability of a user with skill level θ solving item i by

$$p_i(\theta) = c_i + \frac{d_i - c_i}{1 + e^{-a_i(\theta - b_i)}},$$

where a_i, b_i, c_i, d_i are parameters that model discrimination, difficulty, pseudo-guessing, and slip of item i [23].

Another prominent approach is Bayesian Knowledge Tracing (BKT) [27, 42], which uses Markov process to model difficulty of the question items and learning capability of the students. Another well known approach is Deep Knowledge Tracing (DKT) [31], which is the first Deep Learning based KT model (DLKT). Since the introduction of DKT, many researchers have worked on different network structures to capture the complex aspects of the knowledge state. There are a variety of models based on different structures such as DKVMN [43], DKT+ [41], SKVMN [1], SAKT [30], GKT [28], EKT [21], KTM [39], DHKT [40], SAINT [5], AKT [13], and PEBG [22].

While there exist a variety of different KT models, [12] performed a major experiment on the accuracy of three groups of KT models (Markov process, Logistic Regression, Deep Learning) on nine real-world datasets. While deep learning models do show better AUC and RMSE on some datasets, other linear models including the authors' proposed BestLR approach yielded comparable or superior performances on most datasets, which also provided better model interpretability as well.

2.2 Behavioral Testing in Other Applications

To alleviate unexpected behaviors of black-box models, [2] introduces behavioral testing (also known as black-box testing) to test different capabilities of a system in the software engineering perspective. Many studies work on effectively designing test cases [18, 25, 26]. [29] gives a detailed review on the behavioral testing method applied in various software testing domains. In Natural Language Processing, [34] apply the behavioral testing framework to validate the behaviors of general NLP models. They introduce *CheckList*, which is a task agnostic methodology for testing NLP models. *CheckList* is a list of general linguistic capabilities and test type baselines for NLP tasks. It is also a software tool to generate test cases for NLP models.

2.3 Behavioral Studies in Knowledge Tracing

The expected behaviors of the KT models have been discussed in some studies, which point out the adversarial behaviors of KT models and propose new models to alleviate the problem. DKT+ [41] raises two problems of the Deep Knowledge Tracing (DKT) model [31], which are increasing correctness probabilities from false responses, and wavy prediction transition by time. However, these behaviors can naturally occur from the educational effects embedded in the interaction, which we discuss in detail in Section 3.1. The authors add three regularization terms in DKT+ to enhance the consistency of the predictions of DKT, and introduce extra performance measures.

The authors of [19] lists some desirable behaviors based on the monotonicity of the KT models to improve the general ability of the models. Then, they perform three types of novel data augmentation techniques(replacement, insertion, and deletion) and apply them to the training of KT models.

As examined in these relevant studies, the adversarial behaviors and low interpretability of DL models hinders the AIEd society to adopt Deep Learning based KT (DLKT) models and sustain on adopting interpretable models based on BKT, IRT, or Cognitive Diagnosis Models [9]. In this study, we provide a validation framework of DLKT models and conduct an extensive set of experiments on the desired behaviors of DLKT models. Good results highlight the reliability of DLKT models and encourages the model's adoption on most datsets. On the other hand, bad results point out the limitations of DLKT models on some datasets and show spaces for improvement.

3. BEHAVIORAL TESTING FOR KNOWLEDGE TRACING

We propose a black-box behavioral testing framework for knowledge tracing task. First we define the knowledge state (KS), and then elaborate on desirable behaviors of KT models' KS representation. Finally, in Subsection 3.2, we introduce specific experiment setups to assess whether DLKT models satisfy those behaviors.

We define the ***knowledge state*** to be a vector representation of a user's correctness probability on a set of questions Q' at a specific time point. Given the first t interactions $X_t = x_1, x_2, \cdots, x_t$ of a user, we define the user's knowledge state as:

$$KS_t = \left[P(c_t^{(q_j)} = 1 | X_{t-1}, q_j) \right]_{q_j \in Q'} \qquad (1)$$

$c_t^{(q_j)}$ represents the Bernoulli indicator for the event when the user answers correctly to question q_j at time step t, as defined in Table in Section 2.1, which is updated along the provision of the user interaction sequence. Note that a KS is the collection of prediction values of questions, which either is responded or not. We describe the desired aspects of DLKT models in the following section.

3.1 Expected Behaviors of Traced Knowledge State

First we introduce two properties on the ***change*** ΔKS of the knowledge state KS with respect to an atomic change ΔX of the interaction sequence X, which is an insertion or a deletion of single interaction record.

First, **monotonicity** insists that the model's knowledge state should be updated to a more knowledgeable state when the student adds another correctly answered question (positive interaction) or when an interaction record with incorrect response (negative interaction) of the student is deleted. If the Δ is applied in the middle of the interaction record, all changes after the perturbation should hold the property as well. Second, **robustness** insists that a little perturbation in the interaction history should not yield a dramatic change in future knowledge states. The details of the two properties are introduced below.

- **Monotonicity**: If ΔX is a correctly responded interaction $(q_{i_{t_p}}, \mathbf{1})$ at perturbation time t_p, then we can track the relation of $P(c_t^{(q_j)} = 1|X \cup \Delta X, q_j)$ and $P(c_t^{(q_j)} = 1|X, q_j)$ depending on how q_j and ΔX are correlated. In many cases, a positive correlation in correctness probabilities is desired due to the relation of knowledge states:

  $$P(c_t^{(q_j)} = 1|X \cup \Delta X, q_j) > P(c_t^{(q_j)} = 1|X, q_j)$$

  for $t > t_p$ and $q_j \in Q$.

  However, there can also be negatively correlated questions which could be consequences of factors such as limited learning resource. For instance, a college student might sacrifice her studying time on one subject over another when both subjects' examinations are scheduled too closely with each other. This type of circumstance might cause the model to fit a non-monotonic relationship between the two fundamentally unrelated subjects. In most ITS's, however, the target study domain is usually restricted to a single subject, or a set of knowledge components where the student's comprehension on the components is usually positively correlated. Another case is when a negative response increases the correctness probability of a problem as described as an adversarial behavior in [41]. However, the educational effect of consuming a question can give positive feedback on the knowledge state even if the interaction response was wrong. Therefore, we assume the described monotonic behavior in general knowledge tracing environments while simultaneously keeping track of the opposite case in the experiments of Section 4.

- **Robustness**: For any black-box system, it is generally desirable that insignificant change in the system's input leads to limited change in the output. For a knowledge tracing model in an ITS, the input refers to the student's interaction record and the output refers to the model prediction on correctness probability for an encountered question or a set of questions. Therefore, we formulate the robustness of knowledge tracing model as below in a general sense, adopting the ΔX previously defined:

  $$|P(c_t^{(q_j)} = 1|X \cup \Delta X, q_j) - P(c_t^{(q_j)} = 1|X, q_j)| < \epsilon_t$$

  for some ϵ_t, a single interaction ΔX, $t > t_p$, and $q_j \in Q$. If we impose the inequality to always hold on $t = t_p + 1$ and fixed ϵ_1, then it is equivalent to imposing **continuity** on the knowledge state in terms of time-steps. We treat continuity as a specific case of **robustness** and introduce customized test for continuity separately from the test for robustness.

  However, consider a case when q_j and $q_{i_{t_p}}$ in ΔX assess similar concepts, or when the educational effect from the interaction with one question affects the student's correctness probability on the other question. Then an insertion or deletion of one question is prone to have a significant impact upon the predicted correctness probability value of the other for $t > t_p$. Therefore, the defined robustness / continuity need not be universally desirable for all pairs of questions. The impact of this property would eventually depend on the degree of dependency among the questions. Therefore, in the experiments, while assuming robustness for most question pairs, we also carefully track where some questions affect the prediction values of other questions in a notable amount.

Next we discuss what constitutes an expected ***value*** of knowledge state. Testing whether the knowledge state has accurately captured the user's interaction history is in line with the existing quantitative metrics (AUC, ACC) adopted in KT literature. However, the existing test methods focus only on a **single actual question** data provided per each time step whereas we propose to assess knowledge tracing model via its knowledge state on a **virtual question set** in order to provide a more holistic assessment via knowledge state representation.

Although tracing knowledge state on a set of questions would provide a more comprehensive picture of how the user's knowledge is traced, it lacks actual label of correctness on unseen questions at each time step, as discussed in Section 3.1. Therefore, we describe below novel measures to assess correctness of knowledge state under a few purposefully designed circumstances.

- Approximate Label of User-Independent **Initial Knowledge State**: At first prediction step, we approximate correctness label for *all* questions via their 'global difficulty'. The initial knowledge state for all users should represent the model's prior belief of question difficulty before encountering any user-specific interaction record data. It is reasonable to assess the quality of this value in terms of correlation of model's prediction on each question and the question's global difficulty. However, this is an approximation at best since the question's difficulty might not be accurately captured by simple average over its occurrences. If the actual interaction data generated from the ITS provides a very difficult question only after user's knowledge is significantly accumulated, simple average of question correctness labels would not be representative of the question's inherent difficulty. Model's prediction would be high.
- Ideal Value of Knowledge State after **Converged Interaction Data**: We generate obvious edge-case test cases where user's knowledge state on a set of question has converged to a value. We create this virtual dataset by simply repeating an identical interaction record on each question consecutively. The model's prediction value for the question in the repeated interaction should converge to the repeatedly provided label value.
- Approximate Label of Knowledge State in General: It's also possible to approximate a pseudo-label for unseen questions using rolling/expanding averages or IRT-like algorithms which demonstrate more stable and monotonic behavior by design. Although we conjecture such training methodology of DLKT models using pseudo-labels might provide regularization effect, we do *not* include this case in the scope of this work.

Table 1: Behavioral Test Summary

Behavior	Analysis Method
Monotonicity	**Perturbation Test**: Percentage of interaction samples of which model prediction changed in expected direction.
Robustness	**Perturbation Test**: Degree of impact from perturbation across time-steps.
Continuity	**Continuity Test**: Avgerage and maximum change in knowledge state score per step and throughout entire sequence.
Initial Value	**Initial Value Test**: Correlation between question correctness rate and initial knowledge state.
Convergence	**Convergence Test**: Convergence speed as in model AUC and average model output at different time-steps.

3.2 Behavioral Test Setups

Below we describe four behavioral testing setups for DLKT models. First, perturbation tests aim to test model's monotonicity and robustness given an atomic perturbation to the original interaction sequence data. Second, continuity test aims to check whether model's knowledge state representation is continuous along the interaction sequence. Third, initial knowledge state test checks whether the initial knowledge state reflects each question's corresponding difficulty measure. Fourth, convergence test checks whether the knowledge state converges to the expected value and how fast it converges. Following subsections elaborate each of the test setup in further detail. Table 1 provides summary of the tests.

3.2.1 Perturbation Tests

We examine monotonicity and robustness of the model by perturbation tests. We experiment three types of perturbations: insertion, deletion, and flip. For each original interaction sequence, we determine t_p, which is the index of interaction to be perturbed. For the insertion case, we add a new interaction between x_{t_p-1} and x_{t_p}. For the deletion case, we remove the interaction x_{t_p}. For the flip case, we flip the correctness of x_{t_p} from 1 to 0 and from 1 to 0.

In order to check monotonicity, we assess whether the model's predicted correctness probability in the following interaction sequence $X_{[t_p:]}$ changes towards the expected direction. For insertion / deletion / flip of an interaction to which user responded correctly, we examine whether the following future correctness probability $P(c_{t'+1} = 1|X_{t'})$, $\forall t' > t_p$ increases / decreases / decreases, respectively. In the experiments, we fix the perturbation point to be located halfway in the user's original interaction sequence, then measure the proportion of interactions which the model's predicted correctness probability changes towards the expected direction.

To assess the model's robustness, we visualize how the degree of impact from perturbation changes along the time steps from t_p. We expect the degree of impact from perturbation upon the model's prediction to decay gradually as more interactions are fed into the model after the perturbation point t_p.

3.2.2 Continuity Test

We test whether the knowledge state is continuous, in the manner described in the previous section 3.1. For every time-step, we provide the model with not only the original interaction at the corresponding time-step, but also a set of questions Q' simultaneously to construct knowledge state KS_t at the time-step. Although we don't have actual correctness label for those virtual interactions, we only inquire how the knowledge state or the model prediction on Q' evolves along the time-steps.

In the experiments, we approximate a **score** on the user's knowledge state by averaging the model-predicted correctness probability over the sample set of questions Q' to report: (1) average and maximum student score change per single time-step and (2) average student score change and range across 100 time-steps.

3.2.3 Initial Knowledge State Test

Table 2: Dataset Statistics

Dataset	Users	Items	Skills	#Intr.	%Crct.
ASSIST15	14228	100	100	656K	73
ASSIST17	1708	3162	411	935K	37
STATICS	282	1223	98	189K	77
Spanish	182	409	221	579K	77
EdNet-small	5000	13156	118	518K	65
EdNet-med	100000	13518	118	11M	64
EdNet	605763	13528	118	138M	66

We assess the quality of the prior knowledge state embedded by the model by the initial knowledge state test. Without any user-specific record, the prior knowledge state embedded in the model should accurately reflect the average difficulty of the question to all users. Thus, we check Spearman rank correlation and Pearson correlation between the question's average difficulty and the model-predicted prior belief for each question.

In detail, a trained DLKT model M's initial knowledge state for a question q_j can be represented as $P_M(c = 1|\cdot, q_j)$. We compare this with the question correctness rate over the entire dataset as in Eq 2 which is equivalent to the number of correctly responded q_j-interactions over the number of occurrences of q_j based on all user data.

$$\mathrm{gc}_{q_j} = \frac{\sum_{u \in U} |\{x_t^{(u)} | q_{i_t} = q_j, c_t = 1\}|}{\sum_{u \in U} |\{x_t^{(u)} | q_{i_t} = q_j\}|} \quad (2)$$

Consequently, we measure:

$$Corr\Big(\big[P_M(c = 1|\cdot, q_j)\big]_{q_j \in Q}\ ,\ \big[\mathrm{gc}_{q_j}\big]_{q_j \in Q}\Big) \quad (3)$$

This initial knowledge state test pinpoints on whether the learned question embedding in the DLKT model alone has captured any information about the corresponding question's difficulty. Moreover, we emphasize the importance of the initial knowledge state since the state assumed by the model would likely affect the user's first impression on the system to make decisions.

3.2.4 Convergence Test

In convergence test, we assess whether the model's knowledge state value converges to a target value in a desired manner. We generate simple virtual interaction sequence data by repeating an identical interaction for 50 time-steps for each question for both correctness cases. Therefore, the virtual dataset would consist of virtual user interaction sequences of size twice of the number of questions.

In the experiments, we report the model's standard AUC metric at time-steps 5, 10, and 50. We expect significantly high figures as the inquired interaction sequence is extremely simple. We also visualize how the average model prediction value across the questions evolves throughout the 50 time-steps for each of the correctness case. We expect the values to quickly converge to 1 / 0 for interaction sequences of which correctness label is all correct / incorrect, respectively.

4. EXPERIMENTS

In this section, we benchmark three Deep Learning based Knowledge Tracing models DKT, SAKT, and SAINT on the proposed behavioral tests. First, we train optimized models for each architecture-dataset pair by searching hyper-parameters on the train and validation data split. Second, we report the classification accuracy and the AUC metric, which are commonly used for model assessments in the Knowledge Tracing literature. Third, we present the proposed behavioral test results of model instances on well-known datasets for Knowledge Tracing.

4.1 Datasets

We describe the datasets used in our experiments. All datasets are open to the public.

ASSISTments[11] is a dataset containing student interactions from an online tutoring system for solving Massachusetts Comprehensive Assessment System (MCAS) 8th Math test questions. We use the datasets ASSISTments 2015 (**Assistments15**) and ASSISTments Challenge 2017 (**Assistments17**).

STATICS is a dataset containing college student interactions on a one-semester Statics course. This dataset is available in the PSLC DataShop web site [16].

Spanish[20] is a dataset containing middle-school student interaction data for Spanish exercises.

EdNet[6] is the largest public benchmark education dataset containing user interaction data of an online tutoring system, for preparing TOEIC (Test of English for International Communication®). For ablation studies on the size of the dataset, we randomly choose 100,000 users for **EdNet-medium**, and 5,000 users for **EdNet-small**.

Table 3: Model Hyper-parameters

Model	Parameter	Tuning Details
Common	Adam learning rate	0.001, 0.003, 0.01
	Dropout rate	0, 0.25, 0.5
	Embedding dimension	64, 128, 256
	Maximum Seq.Length	100, 200, 400
DKT	# Recurrent Layers	1, 2, 4
SAKT	# Attention Layers	1, 2, 4
SAINT	Warm-up Steps	200, 400, 4000
	# Attention Head	1,4,8

4.2 Models and Algorithms

We perform hyper-parameter tuning on the training of models. For each configuration of hyper-parameters, we choose the model weights with the best validation AUC. In the training step, an early-stopping policy is applied with patience 30, which means that we stop the training process and save the best weights if there is no AUC improvement in the recent 30 validation steps. Among the best weights for each configuration, we choose the weight with the best validation AUC for each dataset, and evaluate the weights with an independent test set for test metrics.

4.2.1 Training Details

In this study, we use DKT, SAKT, SAINT in the experiments . DKT models the student's knowledge state using a Recurrent Neural Network (RNN) by compressing the interaction history in a hidden layer. SAKT is the first KT model that used self-attention layers, where in each layer the question embeddings are queries and interactions embeddings are key and values. SAINT is the first KT model based on Transformers. The sequence of questions is fed into the encoder, and the sequence of responses are fed into the decoder with the encoder output.

Our model hyper-parameters are shown in Table 3. We use the Adam optimizer [15] with default parameters. For SAKT and SAINT, we used the Noam scheme for scheduling the learning rate, and tune the number of warmup-steps.

The original SAKT implementation does not include residual connection from the query. This enforces the first prediction to a same number every time, regardless of the first question provided to the model. Since 3.2.3 becomes redundant, we use the modified SAKT architecture with residual connection. For SAINT and SAKT, the dimension of the feedforward network is set to 4×(model dimension). For SAINT, we use the same number of attention layers for the encoder and the decoder.

4.3 Results: Traditional Assessment

AUC and accuracy results are shown in Tables 4 and 5. The difference of these standard metrics is generally less than 0.01. For KT-based tutoring systems, this difference would be less important than behavioral performance. AUC shows the monotonicity of interactions by all users, and accuracy does not focus on the exact model prediction. On the other hand, behavioral tests can check the performance of model prediction for a single user, and analyze the impact of a single interaction.

Table 4: Standard AUC metric

Model	DKT	SAKT	SAINT
Assistments15	0.7242	0.7226	0.7179
Assistments17	0.7742	0.7650	0.7680
STATICS	0.8269	0.8248	0.8275
Spanish	0.8336	0.8456	0.8364
EdNet-small	0.7332	0.7380	0.7328
EdNet-medium	0.7717	0.7760	0.7722
EdNet	0.7810	0.7905	0.7863
Average	0.7778	0.7804	0.7773

Table 5: Standard Classification Accuracy(%)

Model	DKT	SAKT	SAINT
Assistments15	74.2	74.6	74.4
Assistments17	72.1	71.0	71.8
STATICS	81.4	81.2	81.1
Spanish	81.9	82.6	82.0
EdNet-small	68.2	70.2	69.8
EdNet-medium	72.5	72.6	72.4
EdNet	73.5	74.1	73.9
Average	74.8	75.2	75.1

4.4 Results: Behavioral Testing

4.4.1 Perturbation Tests

We report the test pass rate for insertion, deletion, and flip. The results are shown in Table 6, 7, and 8, respectively. Figure 1 describes the average impact on model prediction from insertion perturbation on each dataset (column) and correctness label of the inserted interaction (row). Figure 2 describes the degree of maximum impact over user sequences from insertion perturbation.

Table 6: Insertion Test Pass Rates(%)

Model	DKT	SAKT	SAINT
Assistments15	70.9	70.3	65.3
Assistments17	69.6	55.7	56.7
STATICS	71.1	61.0	58.2
Spanish	80.1	75.6	60.7
EdNet-small	66.3	78.0	75.9
EdNet-medium	83.2	80.6	77.3
EdNet	72.7	71.6	71.2
Average	73.4	70.4	66.5

Table 7: Deletion Test Pass Rates(%)

Model	DKT	SAKT	SAINT
Assistments15	69.0	66.3	62.1
Assistments17	63.7	54.4	54.6
STATICS	60.7	55.9	49.2
Spanish	81.6	81.9	59.7
EdNet-small	65.6	75.3	71.9
EdNet-medium	80.3	76.8	73.5
EdNet	72.3	68.3	69.5
Average	70.4	68.4	62.9

Table 8: Flip Test Pass Rates(%)

Model	DKT	SAKT	SAINT
Assistments15	77.1	96.3	94.7
Assistments17	69.5	86.1	66.4
STATICS	93.4	92.9	84.7
Spanish	87.5	89.1	83.9
EdNet-small	75.2	95.0	95.8
EdNet-medium	87.8	94.8	95.5
EdNet	79.5	83.6	86.0
Average	81.4	91.1	86.7

- In general, deletion and insertion pass rates range from 60% to 80%, and flip pass rates range from 80% to 90%. Note that a flip can be interpreted as a combination of deletion and insertion. Therefore, the impact of perturbation is supposed to be larger, leading to higher pass rates as compared to insertion/deletion cases. From Figure 1, Figure 6 (Appendix), and Figure 7 (Appendix), we note that the degree of impact from replacement is twice of that from insertion or deletion.

- Robustness: From Figure 1, we observe that the degree of average impact from perturbation gradually decreases along the time-steps in general, and that the average impact is limited by only about 2%. Therefore, the desired robustness holds in terms of average impact.

- Monotonicity: From Figure 1, the average impact from positive/negative perturbation tends to remain posi-

Figure 1: Perturbation Test: Average Impact on Model Prediction from Insertion.

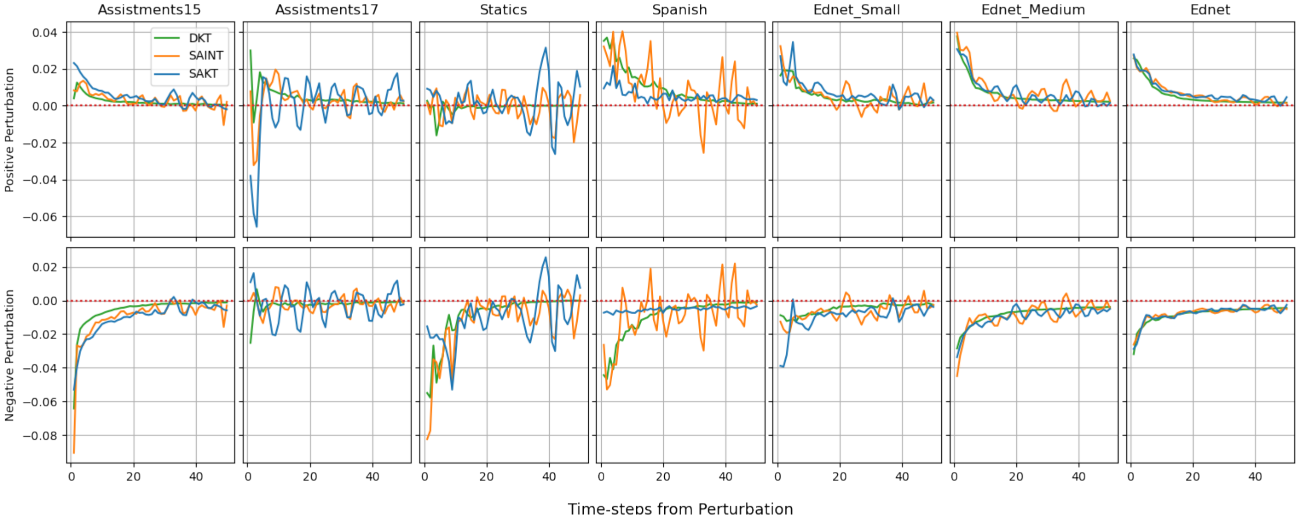
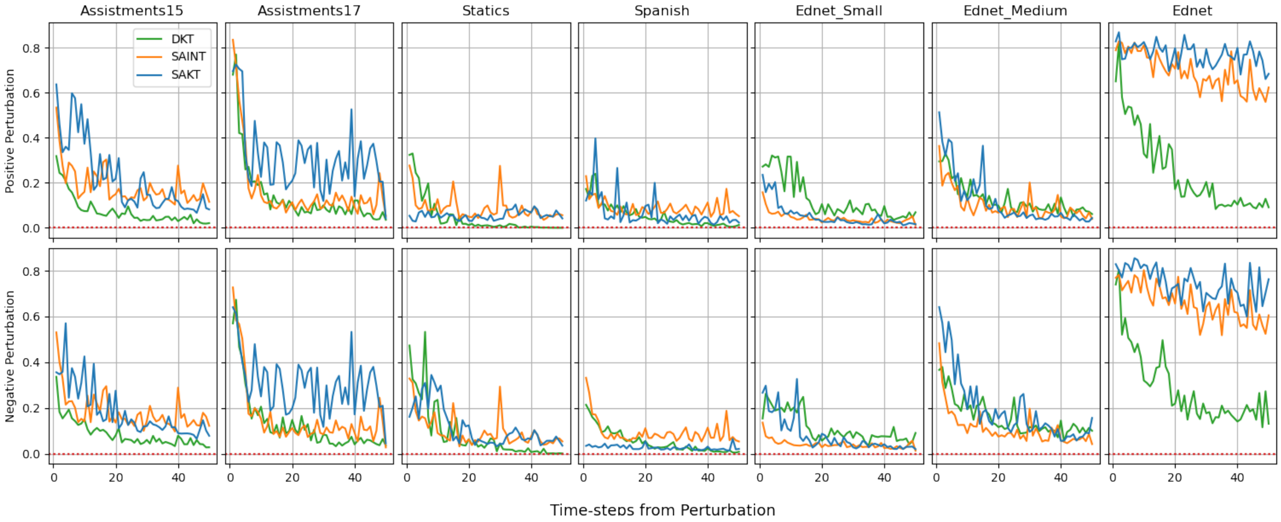


Figure 2: Perturbation Test: Maximum Degree of Impact on Model Prediction from Insertion.



tive/negative, respectively, for Assistments15, EdNet-small, EdNet-medium, and EdNet datasets. On Spanish dataset, such trend is more noisy for SAINT network.

- On Assistments17 and Statics dataset, the expected monotonic behavior from SAKT and SAINT is not observed as the average impact oscillates across zero. This can be also seen from DKT's significantly higher pass rates on the two datasets in Table 6.
- From Figure 2, we note that there exists questions which persist to respond in a larger degree (even up to 80%) after 40 time-steps. Note that on the EdNet dataset, both transformer-based architectures SAKT and SAINT allow larger impacts from perturbations than DKT. This can be explained by the superior performance of the two models on EdNet data over DKT in terms of standard evaluation metrics AUC and ACC.

4.4.2 *Continuity Test*

We report average and maximum step-wise change in KS score over students in Table 9. Apart from the single-step change, we also measure final change of score from the first time-step to the last, and the total range of score explored throughout the time-steps, averaged over all students in Table 10. Sum of absolute change in KS coordinates, or Manhattan distance of KS's along time-steps (averaged over all students) is shown in Figure 3. EdNet-medium was omitted due to its similarity with the plot from EdNet-small.

- Except for Assistments17 and Statics, average score change per single time-step or an interaction remains reasonably low below 5% for all architectures. This suggests that the knowledge state is fairly stable across the time-steps.
- On Assistments17 and Statics, we observe significantly larger changes, especially in DKT. DKT's maximum

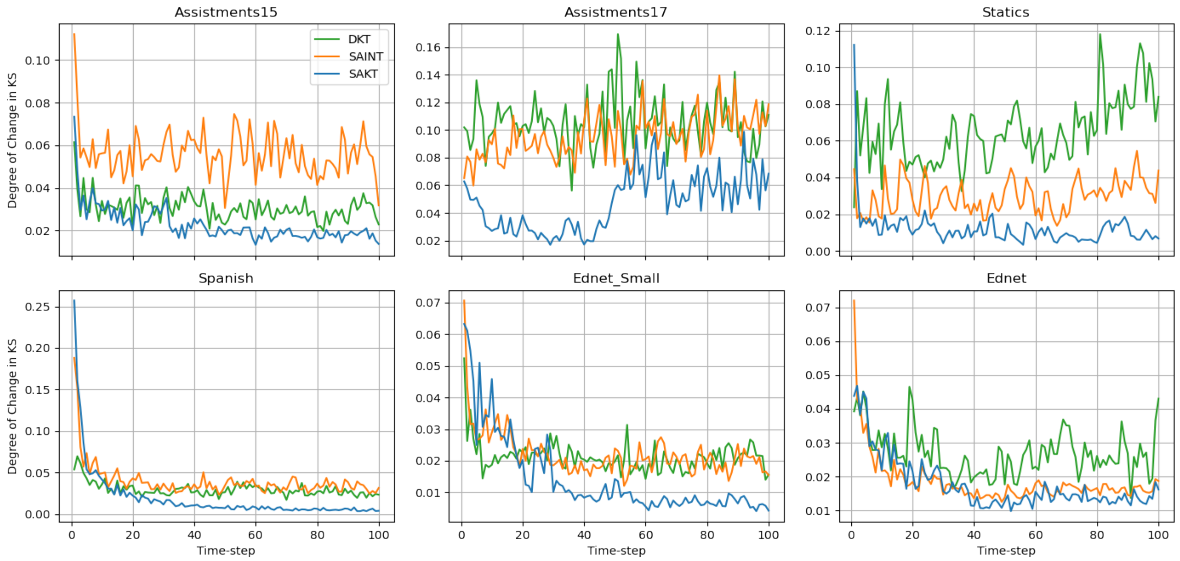


Figure 3: Continuity Test: Average Change in KS along Time-steps

Table 9: Continuity Test: Average / Maximum Student Score Change(%) per Single Time Step

Model		Assist15	Assist17	STATICS	Spanish	EdNet-small	EdNet-med	EdNet	Average
DKT	Avg	2.45	10.48	6.39	2.93	2.04	2.14	2.27	4.10
	Max	16.82	84.97	65.09	20.33	16.98	17.06	33.68	36.42
SAKT	Avg	1.97	3.92	1.15	1.68	1.29	1.19	1.48	1.81
	Max	15.65	56.60	20.24	52.92	21.46	19.21	22.53	29.80
SAINT	Avg	4.64	7.99	2.73	3.48	2.02	1.92	1.29	3.44
	Max	31.80	53.01	24.18	38.74	16.17	22.12	17.19	29.03

Table 10: Continuity Test: Average Student Score Total Change(%) / Total Range(%) over 100 Interactions

Model		Assist15	Assist17	STATICS	Spanish	EdNet-small	EdNet-med	EdNet	Average
DKT	Diff	10.14	11.50	9.94	18.30	9.23	12.82	10.83	11.82
	Range	24.72	63.42	47.54	46.54	22.89	27.41	28.37	37.27
SAKT	Diff	15.96	13.97	15.67	18.36	10.48	13.00	8.95	13.77
	Range	30.98	40.16	22.08	53.33	23.70	23.80	20.69	30.68
SAINT	Diff	13.34	11.62	6.97	20.53	11.32	5.30	9.01	11.15
	Range	37.98	49.96	26.19	51.51	22.98	24.21	20.59	33.35

score change across a single time-step is as high as 85% and 65% for Assistments17 and Statics, respectively.

- In general, we observe decreasing marginal impact of each interaction data as time proceeds.

- From Figure 3 and Table 9, we note that SAKT's knowledge state changes significantly less than other models, consistently throughout all datasets. We also investigated whether this 'speed' of change affects total 'dislocation' of knowledge state in Table 10. Interestingly, SAKT's knowledge state moved by farthest on average (13.77%) while its range explored was the smallest (30.68%) on average. This suggests that SAKT's knowledge state evolution was least volatile.

4.4.3 Initial Knowledge State Test

To assess the validity of initial knowledge states embedded in the model, we measured the correlation of the predicted prior knowledge state and the global question difficulty as described in Section 3.2.3. The results are presented in Table 11. In the scatter plot of Figure 4, we choose the 200 most frequently answered questions from each data-set to show how the initial model predictions and question correctness rates are distributed and correlated.

- We observe from Table 11 that all models' initial knowledge states are positively correlated with the global question difficulty with statistical significance.

- The difference in correlation metrics among datasets is much more significant than that among models.

- Based on the three scatter plots of the first row in the figure, we note that the correlation becomes stronger as the size of dataset grows from EdNet_Small to full EdNet data. Table 11 and Table 2 also suggests that the number of interactions per unique question is positively correlated with the initial knowledge state test metric.

- From the scatter plot, we see that the three models occupy slightly different clustering regions in the plot. For instance, in EdNet_Medium and EdNet dataset, SAINT's initial prediction value is consistently larger than that of the other two models, which suggests ensemble of the models to reduce bias.

Table 11: Initial Knowledege State Test: Correlation(%)

Model	DKT	SAKT	SAINT
Assistments15	85.6	84.8	82.5
Assistments17	63.2	52.2	58.2
STATICS	56.1	58.1	54.6
Spanish	56.5	49.1	44.0
EdNet-small	39.0	38.5	31.6
EdNet-medium	75.1	74.2	74.4
EdNet	87.6	86.5	88.2
Average	66.1	63.3	61.9

4.4.4 Convergence Test

As the dataset we generate and use for the convergence test is extremely simple as described in Section 3.2.4, we expect the KT models' standard performance metrics to increase quickly along the time-steps. For instance, at the fifth time-step, the model would have already received four equivalent interaction record with the same question and the same correctness label for the virtual student. We report the model AUC at time-step 5, 10, and 50 in Table 12. In Figure 3 we also visualize the model's average response across different questions for each correctness label values assumed. We expect the average response plot to quickly converge to either 1 or 0 based on the assumed correctness label value.

- In general, all models show fairly high performance from early time-step of 5, except for Assistments17 dataset.

- Both Table 12 and Figure 5 suggest SAKT consistently achieves fastest convergence to a reasonable value close enough to either 1 or 0. DKT, however, consistently converges to a value farther from the two edges, as compared to the other two models. In particular, for the incorrect case (second row) of the Figure 5, we observe DKT converges to a value higher than 50% (red dotted horizontal line). For the positive case (first row), DKT converges to a correctness probability level around only 70% for Assitments17, EdNet-small, and EdNet-medium.

- DKT's convergence pattern is fairly monotonic while SAKT and SAINT's patterns go through fluctuation which likely pertains to noise.

- It is noteworthy that increasing dataset size from EdNet-small to EdNet-medium and EdNet significantly helps all three models' convergence behavior on both target correctness values, especially for DKT. DKT's convergence value moved signficantly closer to desired values of 1 and 0. For SAKT and SAINT, larger dataset size led to more stable response plot, reducing the degree fluctuation.

- Convergence in the incorrect case and the correct case is asymmetric. While the latter closely achieves the target value of 1, the former case converges around 30% level in most datasets. We attribute this to the tutorial content embedded in each of the interaction, along with the question item used for assessment in the dataset.

4.5 Overview of Experimental Results

Based on the proposed DLKT validation framework, we conducted a comprehensive investigation of three popular DLKT models on seven benchmark datasets to scrutinize the models' behavioral characteristics. The results highlight strengths and weaknesses of three DLKT models. Although DLKT models demonstrated stable and robust behaviors in line with expectation in most datasets, the results revealed few major disadvantages for each models: DKT showed better stability in perturbation tests while the other architectures occasionally presented volatile fluctuation in the response curve. In the continuity test, SAKT presented a significantly smoother evolution of knowledge state, but other models' knowledge state representations were seemingly volatile in a few datasets which strongly precludes DLKT's adoption. On the other hand, this suggests room for improving DLKT models based on the specific issue pinpointed by this framework. For instance, the volatility of KS could be alleviated by direct regularization of the change in the KS. On the other hand, the results from the convergence test showed that DKT was fragile even to simple edge-case data which undermines generalization capability of DKT, as compared to other attention-based architectures.

These behavioral characteristics identified from the proposed framework show that the two popular architectural paradigms, RNN and Attention-based, possess different strengths and weaknesses under KT environment. This also hints that an architectural combination or ensemble approach might alleviate the identified issues to improve both standard KT model evaluation metrics and behavioral characteristic.

5. CONCLUSION

In this work, we introduced the desired properties of knowledge tracing models and proposed a novel model validation framework for Deep Learning based Knowledge Tracing (DLKT) models. Using the framework, we conducted a comprehensive analysis of three popular DLKT models' behavioral characteristics and identify their strengths and weaknesses of the models in seven different benchmark datasets. We believe that the analysis on both strengths and weaknesses diagnosed by the framework would serve as a useful guideline for model enhancement. Also based on the findings from the proposed framework, a customized adoption of DLKT models fitting to the nature of the data and desired behaviors as well as accuracy would become possible.

We believe potential future work includes: (1) tackling the weaknesses of DLKT models identified in this work via architectural modification or model combination, (2) exploring the benefit of data augmentation using virtual edge-case data similar to converging interaction data used in the convergence test, and (3) extending the proposed testing framework beyond the task of knowledge tracing (i.e. student score prediction and item recommendation).

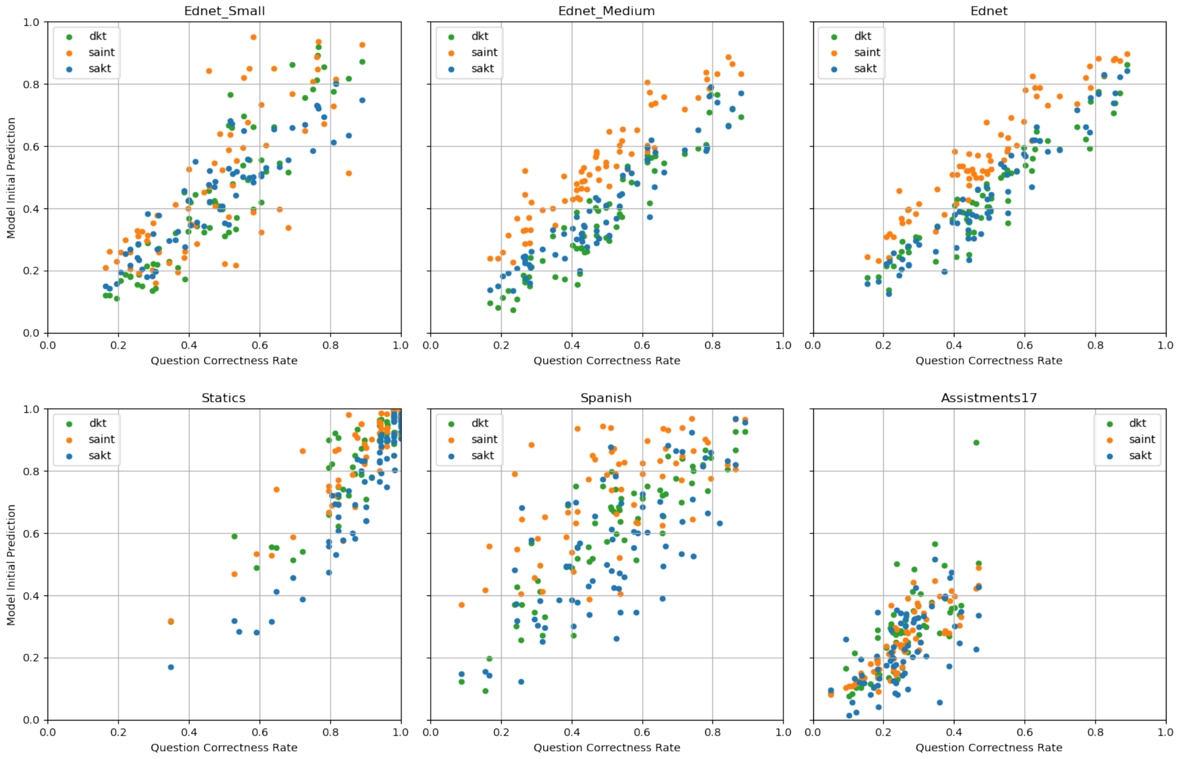


Figure 4: Initial Knowledge State Test Scatter Plot

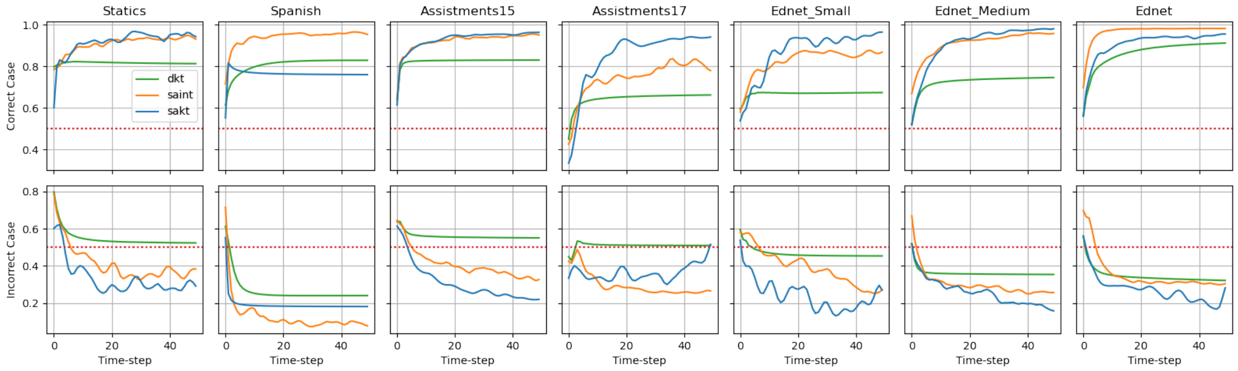


Figure 5: Convergence Test: Average Model Prediction along Converging Interaction Sequence

Table 12: Convergence Test AUC

Step	Model	Assist15	Assist17	STATICS	Spanish	EdNet-small	EdNet-med	EdNet	Average
5	DKT	0.867	0.601	0.829	0.688	0.622	0.790	0.845	0.749
	SAKT	0.885	0.615	0.903	0.805	0.869	0.853	0.861	0.827
	SAINT	0.866	0.609	0.936	0.731	0.628	0.881	0.854	0.786
10	DKT	0.932	0.634	0.924	0.745	0.679	0.874	0.932	0.817
	SAKT	0.961	0.782	0.928	0.923	0.953	0.928	0.951	0.918
	SAINT	0.947	0.763	0.979	0.856	0.757	0.958	0.954	0.888
50	DKT	0.979	0.695	0.983	0.791	0.744	0.954	0.990	0.876
	SAKT	0.998	0.938	0.942	0.993	0.997	0.994	0.995	0.980
	SAINT	0.995	0.939	0.999	0.977	0.963	0.997	0.997	0.981

6. REFERENCES

[1] G. Abdelrahman and Q. Wang. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2019.

[2] B. Beizer. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.

[3] C. G. Brinton and M. Chiang. Mooc performance prediction via clickstream data and social learning networks. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 2299–2307. IEEE, 2015.

[4] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48. IEEE, 2018.

[5] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 341–344, 2020.

[6] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, B. Kim, and Y. Jang. Ednet: A large-scale hierarchical dataset in education, 2019.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[8] R. S. d Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, pages 406–415. Springer, 2008.

[9] J. De La Torre and J. A. Douglas. Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, 69(3):333–353, 2004.

[10] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2013.

[11] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266, 2009.

[12] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *JEDM| Journal of Educational Data Mining*, 12(3):31–54, 2020.

[13] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2330–2339, 2020.

[14] F. Gutierrez and J. Atkinson. Adaptive feedback selection for intelligent tutoring systems. *Expert Systems with Applications*, 38(5):6146–6152, 2011.

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

[16] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43:43–56, 2010.

[17] E. R. Lai. Metacognition: A literature review. *Always learning: Pearson research report*, 24:1–40, 2011.

[18] M. Last, S. Eyal, and A. Kandel. Effective black-box testing with genetic algorithms. In *Haifa Verification Conference*, pages 134–148. Springer, 2005.

[19] S. Lee, Y. Choi, J. Park, B. Kim, and J. Shin. Consistency and monotonicity regularization for neural knowledge tracing, 2021.

[20] R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[21] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[22] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu. Improving knowledge tracing via pre-training question embeddings. *arXiv preprint arXiv:2012.05031*, 2020.

[23] E. Loken and K. L. Rulison. Estimation of a four-parameter item response theory model. *British Journal of Mathematical and Statistical Psychology*, 63(3):509–525, 2010.

[24] M. I. Lopez, J. M. Luna, C. Romero, and S. Ventura. Classification via clustering for predicting final marks based on student participation in forums. *International Educational Data Mining Society*, 2012.

[25] Y. K. Malaiya. Antirandom testing: Getting the most out of black-box testing. In *Proceedings of Sixth International Symposium on Software Reliability Engineering. ISSRE'95*, pages 86–95. IEEE, 1995.

[26] L. Mariani, M. Pezze, O. Riganelli, and M. Santoro. Autoblacktest: Automatic black-box testing of interactive applications. In *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, pages 81–90. IEEE, 2012.

[27] J. Martin and K. VanLehn. Student assessment using bayesian nets. *International Journal of Human-Computer Studies*, 42(6):575–591, 1995.

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

[29] S. Nidhra and J. Dondeti. Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, 2(2):29–50, 2012.

[30] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[31] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[32] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.

[33] Z. Ren, X. Ning, A. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*. ERIC, 2019.

[34] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.

[35] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, 68:458–472, 2013.

[36] H. Tan, J. Guo, and Y. Li. E-learning recommendation system. In *2008 International Conference on Computer Science and Software Engineering*, volume 5, pages 430–433. IEEE, 2008.

[37] D. Thissen and M. Orlando. Item response theory for items scored in two categories. 2001.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[39] J.-J. Vie and H. Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.

[40] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, 2019.

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

[42] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.

[43] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774, 2017.

APPENDIX

Figure 6: Perturbation Test: Average Impact on Model Prediction from Deletion.

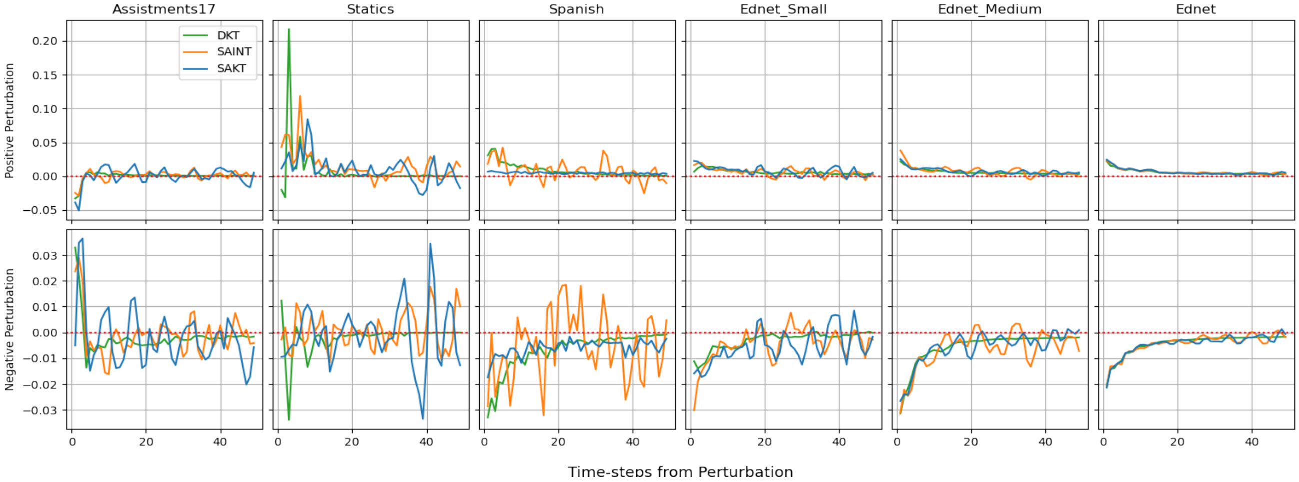
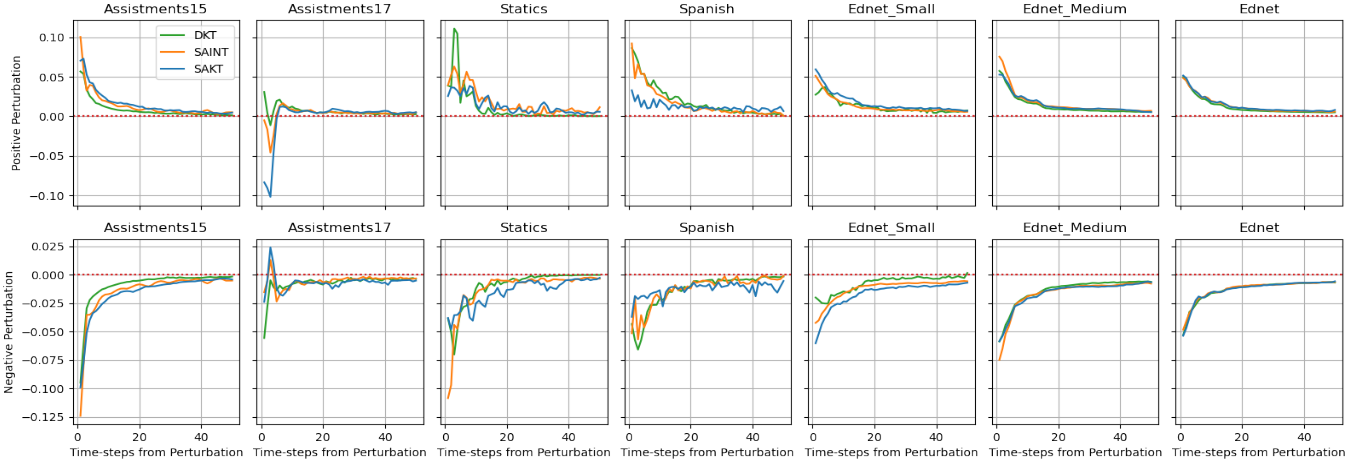


Figure 7: Perturbation Test: Average Impact on Model Prediction from Flip.


Assistments15
Assistments17
Statics
Spanish
Ednet_Small
Ednet_Medium
Ednet
DKT
SAINT
SAKT
Positive Perturbation
Negative Perturbation
Time-steps from Perturbation