Self-Improvement for Computerized Adaptive Testing

Yannick Rudolph* (✉), Kai Neubauer*, and Ulf Brefeld

Leuphana University, 21335 Lüneburg, Germany
{yannick.rudolph,kai.neubauer,ulf.brefeld}@leuphana.de

Abstract. Computerized adaptive testing (CAT) allows for assessing latent traits and abilities of students with fewer items and in less time due to an individualized item selection algorithm based on previous responses. Following recent machine learning solutions to CAT, we study learning both the underlying response model for cognitive diagnosis and a policy for the item selection algorithm jointly from offline training data. While the task of the response model is to predict performances on all unseen items for a user, the goal of the policy is to select the subset of items which maximizes information for the response model. Since subset selection is a combinatorial problem, we propose to leverage an iterative self-improvement approach to policy learning from the field of neural combinatorial optimization while accounting for interdependencies between response model and policy. We specifically focus on the generalization capabilities of transformer-based models and, in contrast to related work, do not rely on optimization of local variables during inference. We report on empirical results.

Keywords: Educational data mining · Computerized adaptive testing · Neural combinatorial optimization · Self-improvement.

1 Introduction

Classical test theory [37,26] focuses on estimating latent traits and abilities of students by observing their responses in tests. In computerized adaptive testing (CAT), questions are adaptively selected according to a student's performance on previously seen items. Due to this personalization, selected questions are on average better suited to assess latent traits compared to classical test theory, rendering adaptive tests more accurate and shorter in terms of test length and time [43,42]. The theoretical foundation of CAT is closely related to item response theory (IRT) [23,29,19] that introduces a large family of models to estimate latent traits of students. Traditional CAT approaches use IRT-based models to estimate latent abilities of students and difficulties of questions to predict future responses on unseen questions and to guide item selection.

Recent machine learning approaches to CAT proposed to employ learned policies for question selection [10,51] and have framed CAT as an iterative

* Authors contributed equally to this work.

subset selection problem [52] while deep learning architectures such as the neural cognitive diagnosis model (CDM) [45] have been proposed to replace classical IRT-based response models.

In this paper, we leverage the observation that the CAT setting is related to problem settings and solutions from the field of neural combinatorial optimization (NCO), where NCO is concerned with learning policies to obtain generalizable solutions to combinatorial problems [16]. Specifically, we propose novel response and policy models that build upon transformer- [40] and NCO-architectures [41,2], and adapt a self-improvement training approach to policy learning for NCO [28] for CAT.

The remainder is structured as follows: Section 2 introduces the problem setting. We present our main contribution in Section 3 and report on experiments in Section 4. Section 5 briefly reviews related work, and Section 6 concludes.

2 Preliminaries

The goal in computerized adaptive testing (CAT) is to assess latent traits and abilities of students as accurately as possible by individually selecting questions for every student. CAT models thus consist of (i) a *response model* for cognitive diagnosis to estimate student abilities and (ii) a *policy* that iteratively adapts the selection of questions to a given student. In practice, both models are trained on the outcomes of a calibration pre-study and then applied to test unseen students.

Given a calibration study where N students answered K questions.[1] The calibration study is represented by N sets $\mathcal{Q}^n$, $1 \leq n \leq N$, of cardinality K. Every element of $\mathcal{Q}^n$ is a tuple (q_k^n, r_k^n), consisting of the k-th question that has been answered by the n-th student. The binary response variable r_k^n indicates whether her answer was correct ($r_k^n = 1$) or incorrect ($r_k^n = 0$). In the remainder, $\mathcal{Q}^n$ is also referred to as the *question bank* of the n-th student.

Similar to [52], we phrase learning the policy as a subset selection task: For the n-th student, we *iteratively* aim to select the subset $S \subset \mathcal{Q}^n$ with $|S| = T \leq K$ questions that provides maximal information about her latent ability θ^n_{true}. Note that selecting the subset iteratively is key to the CAT setting as this allows us to update the response model with observed responses. Initializing the subset at $t = 0$ with $S_0^n = \emptyset$, we proceed as follows. At time $0 < t \leq T$, we have selected t questions $S_t^n = \{(q_1^n, r_1^n), ..., (q_t^n, r_t^n)\}$.[2] A straightforward way to learn the latent

[1] Without loss of generality, we do not assume that all students answer the same K questions; every student may actually have answered a different number of questions. To not clutter notation, we ignore extra indices and write K for all students.

[2] The selection of questions induces a partial permutation of elements in $\mathcal{Q}^n$ forming the set S_t^n. As above, we prefer not to clutter notation and refrain from defining a proper permutation operator and simply enumerate the tuples by their time index t. That is, we lose a clear identifier and simply consider elements $(q_j, r_j) \in S_t^n$ and $(q_j, r_j) \in \mathcal{Q}^n$ as being different. It should be clear from the context whether we refer to the numbering in S_t^n or $\mathcal{Q}^n$.

ability θ_t^n of the n-th student is offered by minimizing

$$\theta_t^n = \arg\min_{\theta^n} \sum_{(q,r)\in S_t^n} \ell(r, p_\psi(q;\theta^n)), \tag{1}$$

where $\ell(\cdot,\cdot)$ is an appropriate loss (e.g., binary cross entropy), and $p_\psi(\cdot)$ a response model estimating the probability of a correct response, e.g., a neural cognitive diagnosis [10] or Rasch model [29], with global parameters ψ. In practice, these parameters are often fixed and, for example, correspond to item difficulties in IRT [29]. The underlying assumption is that a well-estimated response model is able to predict student performance accurately, resulting in a small loss on the subset S_t^n.

Our approach grounds on the idea that the performance of the n-th student on question bank $\mathcal{Q}^n$ serves as a good proxy for her true but unknown latent abilities θ_{true}^n. This can be shown in the limit by assuming a consistent estimator of latent abilities and a question bank with infinite cardinality [52]. Hence, given a learnable policy π_ϕ with parameters ϕ, we formalize our selection algorithm as

$$q_{t+1}^n \sim \pi_\phi(\theta_t^n; \mathcal{Q}^n \backslash S_t^n, p_\psi),$$

and minimize the empirical risk jointly over parameters ϕ and ψ on so far *unused* (i.e., not yet contained in S_t^n) elements of the question bank $\mathcal{Q}^n$. We arrive at the following optimization problem,

$$\begin{aligned} \min_{\psi,\phi} \quad & \frac{1}{N}\sum_n \sum_{(q,r)\in\mathcal{Q}^n\backslash S_t^n} \ell(r, p_\psi(q|\theta_t^n)) \\ s.t. \quad & \theta_t^n = \arg\min_{\theta^n} \sum_{(q,r)\in S_t^n} \ell(r, p_\psi(q;\theta^n)), \end{aligned} \tag{2}$$

where subsets S_t^n are sampled autoregressively from policy π_ϕ.

An alternative to learning a policy has been introduced by uncertainty sampling [18,33]. The idea of uncertainty sampling is to select at every time the question that the response model is most uncertain about. That is, at time $t+1$, uncertainty sampling chooses the question q for which holds

$$q = \arg\min_{q_m} |p_\psi(q_m|\theta_t^n) - 0.5|. \tag{3}$$

This strategy relies on a well-calibrated response model. Proposition 1 shows that uncertainty sampling is optimal if the response model is optimal as well.

Proposition 1. *A perfectly calibrated response model for which*

$$p_\psi(q|\theta_{t_0}^n) = \mathbb{E}\left[r|q, \theta_{\text{true}}^n\right]$$

holds for all $(q,r) \in \mathcal{Q}^n$ *and* $1 \le n \le N$ *renders the* uncertainty baseline *in Equation (3) an optimal policy given the learning task as introduced above.*

Proof. Assume the response model $p_\psi(q|\theta_t^n)$ converges after an update $t_0 \in \{1, \ldots, T\}$ to the optimal model for student n and the remaining items of the question bank $\mathcal{Q}^n \backslash S_{t_0}^n$. This directly implies that $\forall\,(q,r) \in \mathcal{Q}^n : p_\psi(q|\theta_{t_0}^n) = p_\psi(q|\theta_{t \geq t_0}^n)$ and hence also $p_\psi(q|\theta_{t \geq t_0}^n) \equiv \mathbb{E}\,[r|q, \theta_{\text{true}}^n]$. Thus, for all $t \geq t_0$, uncertainty sampling in Equation (3) trivially picks the question that leads to the largest minimization of the loss in Equation (2) as all other remaining questions in $\mathcal{Q}^n \backslash S_{t \geq t_0}^n$ are closer to the expected response and realize smaller losses in expectation. In turn, the policy is optimal for $\mathcal{Q}^n \backslash S_{t \geq t_0}^n$. □

Uncertainty sampling is a greedy heuristic and can lead to suboptimal results in practice. While the proposition above provides a motivation for its application, in practice the learned response model will hardly be perfect due to noisy data (e.g., guessing and slip probabilities or miscalibration). Conditioning the response model on previously seen question-response tuples in S_t^n should increase the quality of the model, since asking more questions should lead to an improvement of predictive accuracies. That is, we have $p_\psi(q|S_t^n) \neq p_\psi(q|S_{t+1}^n)$ in general.

3 Toward self-improvement training for CAT

We now propose a deep learning approach that relies on strong generalization properties of modern deep learning architectures to solve the optimization problem in Equation (2). Specifically, we develop novel response and policy models for CAT that leverage the encoder-decoder structure of transformers. Further, we adapt self-improvement training [28] to learn the policy, leveraging similarities between our subset selection problem and neural combinatorial optimization.

3.1 Amortized student representation

Instead of local variable optimization per student, we consider the student representation θ_t^n to be an encoded representation of S_t^n given by

$$\theta_t^n = p_\psi^{\text{enc}}(S_t^n),$$

as obtained by a transformer encoder-like self-attention architecture. The encoder $p_\psi^{\text{enc}}(S_t^n)$ itself is part of a response model $p_\psi(q|p_\psi^{\text{enc}}(S_t^n)) = p_\psi(q|S_t^n)$ that operates on the observed subsets S_t^n of questions and responses for student n at time t. We argue as follows: if we can learn the encoder on calibration data such that it generalizes well on unseen examples, we may discard local parameter optimization per student since all relevant individual traits are already captured by the encoded representations. Additionally, we can also base the policy model on the encoded student representation, with parameters either optimized independently or shared with the encoder. Thus, incorporating an encoder-like self-attention architecture allows us to discard learning the latent abilities via Equation (1) and simplify the optimization problem in Equation (2) as

$$\min_{\psi,\phi} \frac{1}{N} \sum_n \sum_{(q,r) \in \mathcal{Q}^n \backslash S_t^n} \ell(r, p_\psi(q|S_t^n)), \tag{4}$$

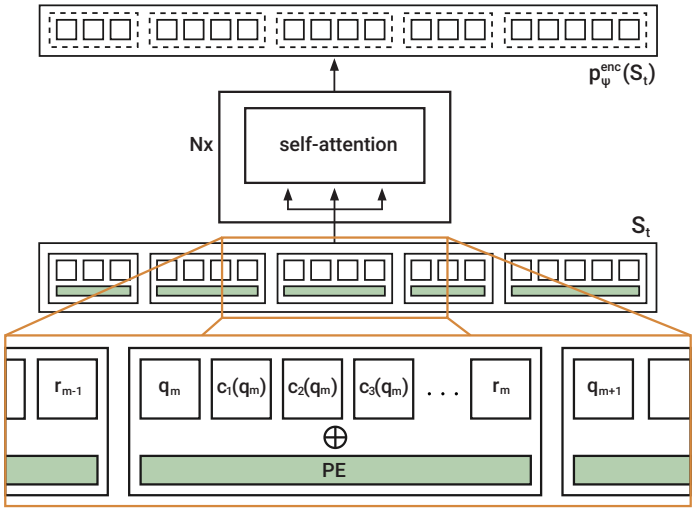

Fig. 1: Sketch of our student representation, where $c_i(q)$ denote question features

where subsets S_t^n of questions-response tuples for student n are sampled from policy π_ϕ and ψ and ϕ are global parameters as before. Trusting global parameters and learned representations to be sufficient for generalization resembles the idea of *amortized inference* [49] prominent in variational autoencoders [14,30].

In comparison to classical IRT models [29], however, we lose interpretability by discarding the explicit representation of the latent student traits θ_{true}^n. Instead, we trade interpretability for a better response model. In cases where interpretability is necessary, we could either train additional response models or apply post-hoc and model-agnostic strategies [8,31]. We consider both ideas straightforward additions to our contribution but focus on the predictive performance that is integral to personalization in CAT at test time.

Another advantage of our approach is the ability to process possibly rich feature representations of questions and responses. Recall that the input to the student encoder is the set S_t^n of question-response tuples which could be represented in the form of sets of features. The encoder then operates on a *set of sets*, given that the features are discretely tokenizable. To showcase the benefit of this extension, we experimentally include knowledge components that describe skills that are necessary to solve a particular question as part of a domain model. Our approach allows to include any number of descriptive features for questions, responses, and also students.

Since self-attention is position-agnostic, we apply positional encoding (PE) to link question-response tuples to the tokenized features (here and elsewhere we apply PE by element-wise addition denoted by $\oplus$). In addition, we include learnable *start tokens* that enable us to learn a student representation for S_0. The encoder architecture follows a stack of standard transformer encoder layers [40] comprised of multi-head self-attention, residual connections [11], dropout

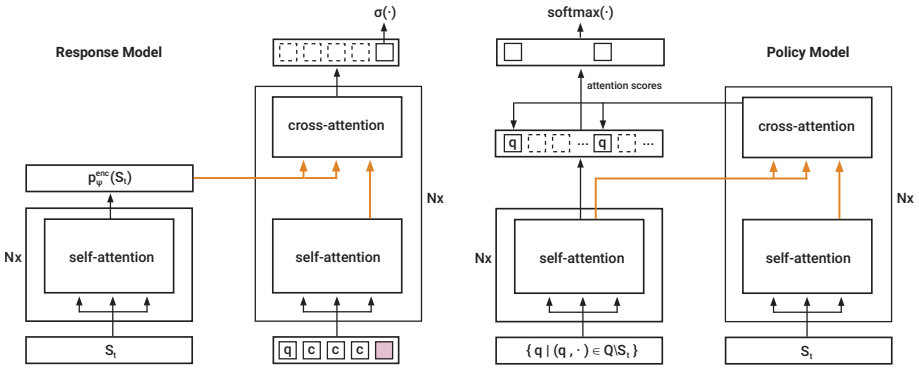

Fig. 2: Sketch of proposed response and policy models

[38], layer normalization [1], and feed-forward neural networks. Since attention in the transformer architecture scales quadratically [15], our student representation can be computed in $\mathcal{O}((f_q + f_r + f_n)^2)$ where $f_{(\cdot)}$ denotes the number of tokens in question (f_q), response (f_r), and student representations (f_n) up to time t. Figure 1 visualizes the model for the student representation.

3.2 Response and policy models

The response and policy models are both based on standard transformer encoder-decoder architectures [40] where only positional encoding, problem specific masking in self- and cross-attention and classifier architectures are adjusted. We present both models in the following sections; see Figure 2 for a visualization of the two architectures.

Response model The task of the response model is to estimate the probability that a student answers a question correctly. The model is conditioned on all previously observed question-response tuples, so that, after having asked t questions, we can estimate the probability $\hat{r} = p_\psi(q|p_\psi^{\text{enc}}(S_t))$, for every unseen question $q \in \{q|(q,\cdot) \in \mathcal{Q}\backslash S_t\}$.

To compute these probabilities, we pass the set of features for a desired question q as well as a learnable *query token* into a transformer decoder, where the self-attention layers operate on question features and the query token. The cross-attention layers attend from the decoder question representation to the encoded student representation. The response can be predicted by applying a binary classifier to the transformed representation of the query token.

Different from standard decoder layers of the transformer architecture, there is no masking involved, as all tokens representing question q are allowed to attend to the full representation as obtained via $p_\psi^{\text{enc}}(S_t)$. During training and inference we repeat the encoded student representation $p_\psi^{\text{enc}}(S_t^n)$ for each question in $\mathcal{Q}^n\backslash S_t^n$ for efficiency. At time t, self-attention in the decoder scales with $\mathcal{O}\left(f_q^2\right)$, while

cross-attention scales with $\mathcal{O}\left(f_{S_t} \times f_q\right)$, where f_{S_t} extracts the number of tokens in the subset S_t; it usually holds that $f_q < f_{S_{t>0}}$.

Policy model The encoder of the policy model transforms the set of sets of tokenized features of available questions $q \in \{q|(q,\cdot) \in \mathcal{Q}\}$, where we apply positional encoding to inform the transformer about the association between tokens and questions, i.e. we apply the same positional information to tokens from the same interaction. The student representation as presented in Section 3.1 results from the application of self-attention in the decoder, which operates on S_t ($\oplus$ PE). In the cross-attention layers, student representations obtained via self-attention on S_t attend to the candidate questions $q \in \{q|(q,\cdot) \in \mathcal{Q} \backslash S_t\}$ as transformed by the encoder. To obtain a probability distribution over all candidate questions at timestep $t+1$, we apply a softmax to the final cross-attention scores.[3] To efficiently train the model via teacher forcing, we train on all $q \in \{q|(q,\cdot) \in \mathcal{Q}\}$ and the complete subset S_T, which we treat as a sequence of question-response tuples with appropriate autoregressive block-structured masking applied in the self-attention (where all tokens from tuple (q_t, r_t) can only attend to tokens from $S_{\leq t}$). In the cross-attention, we mask out all question tokens already included in S_t. Self-attention in the encoder scales with $\mathcal{O}(f^2_{|\{q|(q,\cdot)\in\mathcal{Q}\}|})$, self-attention in the decoder with $\mathcal{O}(f^2_{S_T})$, while cross-attention scales with $\mathcal{O}(f_{S_T} \times f_{\{q|(q,\cdot)\in\mathcal{Q}\}})$. Within our approach this renders the policy model the component with the highest computational and memory complexity.

Since response and policy models operate on either the same or similar tokens, we experimented with weight sharing within and between both models. We finally settled on sharing weights in the encoder and decoder for self-attention in both, response and policy model (cf. [32]) but dropped sharing parameters between response and policy model for a lack of noticeable improvements.

3.3 Self-improvement training

In this section, we describe a method to train the policy model on previously recorded calibration data. Instead of applying reinforcement learning, we choose to adapt a recent self-improvement training approach proposed for neural combinatorial optimization [28] and language models [12]. The idea of this strategy is to train a model on its own output in a supervised fashion.

Given a trained response model, we sample multiple subsets S_T^n for every student from a randomly initialized policy model. To induce a learning signal, we focus for each student on the sampled subset that achieves the lowest average binary-cross entropy loss after predicting response probabilities for all unseen questions. The *best* subsets are now used as target sequences for training the policy model in a supervised fashion. When the performance of the policy improves (as measured by the accuracy of the response model with subsets sampled greedily from the policy), the response model is finetuned on subsets sampled from the

[3] We average over attention heads in our implementation.

Algorithm 1 Self-improvement training for CAT

Require: Offline train and validation data $\mathcal{D}_{\text{train}} = \{\mathcal{Q}^n\}_{n=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{val}} = \{\mathcal{Q}^n\}_{n=1}^{N_{\text{val}}}$
Require: Response model p_ψ pretrained on random subsets S_T
Randomly initialize policy π_ϕ and set $\pi_{\text{best}} \leftarrow \pi_\phi$
for each epoch **do**
  for each $n \in \{1, ..., N_{\text{train}}\}$ **do**
    Sample m subset candidates $S^n_{\text{candidates}} := \{S_T^{n,1}, \dots, S_T^{n,m}\} \sim \pi_{\text{best}}$
    Set $S_T^n = \arg\min_{S_T \in S^n_{\text{candidates}}} \sum_{(q,r) \in \mathcal{Q}^n \setminus S_T} \ell(r, p_\psi(q|S_T))$
  end for
  for each batch **do**
    Sample tuples $(\mathcal{Q}^n, S_T^n)$ of question bank and corresponding subset
    Update π_ϕ with gradient optimizing next-step prediction
  end for
  if greedy performance of π_ϕ on $\mathcal{D}_{\text{val}}$ better than π_{best} **then**
    $\pi_{\text{best}} \leftarrow \pi_\phi$
    Finetune p_ψ on $S_T^n \sim \pi_{\text{best}}$
  end if
end for

new policy. We then continue to sample subsets from the best policy to train both the policy and finetune the response model in an alternating fashion.

Algorithm 1 sketches the complete training loop. In contrast to [28], we do not have a fixed objective function but alternate between training policy and response model. This scheme is motivated by policy-induced shifts in the distribution of subsets S_T selected during adaptive testing, which can be accounted for by finetuning the response model by training on subsets S_T selected by the policy. We alternate between training both models, since finetuning the response model will in principle affect the optimal policy. The initial response model is trained on randomly sampled subsets. We use early stopping on validation data for training and finetuning the response model as well as for training the policy.

Neural combinatorial optimization deals for example with synthetic traveling salesman instances and millions of training examples [28]. By contrast, calibration studies in CAT are usually orders of magnitude smaller than that. Thus, we need to adapt the strategies used in neural combinatorial optimization to cope with overfitting and small sample sizes. During training on all student question banks, we sample sequences anew in every epoch while [28] reuse sampled sequences. We also optimize the policy with respect to the complete subset instead of optimizing a single timestep as is done in [28], who argue in favor of a more expressive model that does not support teacher forcing.

4 Experiments

In this section, we empirically evaluate our approach on real world data, support the design choices in our student representation and response model on a stan-

dardized benchmark, and shed light on the interplay between response model, learned policies and the uncertainty policy on artificially generated data.

4.1 Computerized adaptive testing

We evaluate the performance of our approach on a real CAT problem from the NeurIPS 2020 Education Challenge [46]. We train, validate and test on all 6148 available students in 5-folds (training on 60% of the students, using different 20% for validation and testing in each fold). Following related work from neural combinatorial optimization, we restrict the question bank of each student to contain maximally 100 questions each, covering all 948 questions in the dataset.

In absence of the true latent abilities of the involved students, we need to resort to a proxy for a quantitative evaluation and compute predictive performances on unseen test questions instead [10,52]. In our evaluation we resort to calculate mean accuracy and AUC for N^{test} students who are not present in the training data using questions given by $\mathcal{Q}^n \backslash S_T^n$. This out-of-sample evaluation directly addresses the desired goal of having every student answer every question in her question bank, but in practice being able to ask her only T questions and estimating the remaining responses as best as possible.

We compare three versions of our model: a random question selection and an uncertainty sampling policy, both using the learned response model, as well as the full model trained with self-improvement. We further compare against an implementation of BECAT [52][4], applied to a standard IRT [23,29] model, as well as baseline policies based on maximum Fisher and Kullback-Leibler information [24,3]. Our implementations build upon the code provided by [19][5]. The neural CDM [45] did not achieve comparable performances and stayed significantly below the results of the IRT model with a random policy. This is most likely caused by difficulties in optimizing the neural CDM itself. In the experiments with our approach we relied on standard ancestral sampling to optimize policy and response models. Further implementation details are provided in our code repository for this paper.[6]

Tables 1 and 2 show the resulting accuracies and AUCs for test lengths of $T \in \{5, 10, 20\}$, following the experiment protocol in [52].[7] Compared to adaptive question selection with BECAT, our policy model enables significantly faster inference by a factor of over 20. Accuracy and AUC results show that methods building upon our student representation and response model generally outperform the baseline policies with an IRT model. Overall, uncertainty sampling with our response model performs best. Although self-improvement training achieves comparable performance on a test length of 5 the uncertainty sampling

[4] BECAT's results are significantly better than other recent CAT approaches [10,51], especially on longer test lengths, according to [52].

[5] Code available at https://github.com/bigdata-ustc/EduCAT.

[6] Experiments in this paper can be reproduced with code and hyperparameters provided at https://github.com/kainbr/cat_self_improvement.

[7] Our experiments differ from [52] and results are not directly comparable.

Table 1: Accuracies on the NeurIPS 2020 Education Challenge. Markers $*$, $\circ$ and $\bullet$ indicate whether our method with self-improvement is statistically superior, equal or inferior to baselines, using a paired t-test at the 0.01 significance level.

Metric@Step	Accuracy@5	Accuracy@10	Accuracy@20
IRT w/ Random	0.6538 ± 0.0047 $*$	0.6693 ± 0.0063 $*$	0.6798 ± 0.0089 $*$
IRT w/ MFI	0.6669 ± 0.0021 $*$	0.6815 ± 0.0015 $*$	0.6915 ± 0.0026 $*$
IRT w/ KLI	0.6626 ± 0.0016 $*$	0.6787 ± 0.0011 $*$	0.6891 ± 0.0021 $\circ$
IRT w/ BECAT	0.6533 ± 0.0010 $*$	0.6727 ± 0.0031 $*$	0.6885 ± 0.0017 $*$
Our w/ Random	0.6649 ± 0.0023 $*$	0.6781 ± 0.0047 $\circ$	0.6912 ± 0.0019 $*$
Our w/ Uncertainty	**0.6758** ± 0.0034 $\circ$	**0.6959** ± 0.0033 $\bullet$	**0.7239** ± 0.0047 $\bullet$
Our w/ Self-Improv.	<u>0.6753</u> ± 0.0030	<u>0.6855</u> ± 0.0027	<u>0.6967</u> ± 0.0034

Table 2: AUC results on the NeurIPS 2020 Education Challenge.

Metric@Step	AUC@5	AUC@10	AUC@20
IRT w/ Random	0.7128 ± 0.0049 $*$	0.7304 ± 0.0068 $*$	0.7403 ± 0.0093 $*$
IRT w/ MFI	0.7273 ± 0.0022 $*$	0.7466 ± 0.0022 $\circ$	0.7615 ± 0.0032 $\circ$
IRT w/ KLI	0.7235 ± 0.0013 $*$	0.7443 ± 0.0011 $\circ$	0.7597 ± 0.0016 $\circ$
IRT w/ BECAT	0.7116 ± 0.0017 $*$	0.7363 ± 0.0030 $*$	0.7547 ± 0.0028 $\circ$
Our w/ Random	0.7256 ± 0.0019 $*$	0.7446 ± 0.0026 $*$	0.7576 ± 0.0025 $\circ$
Our w/ Uncertainty	<u>0.7363</u> ± 0.0039 $\circ$	**0.7589** ± 0.0037 $\bullet$	**0.7843** ± 0.0043 $\bullet$
Our w/ Self-Improv.	**0.7372** ± 0.0032	<u>0.7490</u> ± 0.0035	<u>0.7637</u> ± 0.0041

policy performs better with longer test lengths. Under the assumption that our response model performs well, the results are in line with the intuition provided by Proposition 1. We will address this observation again in Section 4.4.

4.2 Performance of the response model

In this section, we focus on the evaluation of the student representation and response model on a related educational task, namely knowledge tracing (KT, [7]). The task in KT is to predict binary responses of a student interacting sequentially with an intelligent tutoring system: After observing t question-response tuples, we aim to predict the student's response for question q_{t+1}. A key difference to CAT is the assumption that the knowledge of a student may change over time. Nevertheless, knowledge tracing constitutes a sequential prediction task where a response model is learned on a fixed and fully observable policy.

Differences to the CAT task are for example that question-response tuples in the student representation are treated as a sequence instead of a set. We thus need to adapt the handling of positional encoding which should now reflect temporal relations. To that end, we employ a rotary embedding [39], which is useful in conveying relational positional information to the transformer architecture. For efficiency, we employ an encoder only architecture with an autoregressive mask applied to the student representation, such that tokens corresponding to

Table 3: Results for the knowledge tracing task on the Ednet dataset. Markers $*$, $\circ$ and $\bullet$ indicate whether our method is statistically superior, equal or inferior to baselines, respectively, using a paired t-test at the 0.01 significance level.

	AUC	Accuracy
simpleKT [20]	0.6593 ± 0.0041 $*$	0.6565 ± 0.0029 $*$
SAINT [5]	0.6598 ± 0.0023 $*$	0.6511 ± 0.0039 $*$
AKT [9]	0.6705 ± 0.0024 $*$	0.6645 ± 0.0035 $*$
DTransformer [48]	0.6719 ± 0.0037 $*$	0.6656 ± 0.0032 $*$
FoLiBiKT [13]	0.6721 ± 0.0018 $*$	0.6666 ± 0.0028 $*$
DIMKT [34]	0.6748 ± 0.0030 $*$	0.6700 ± 0.0038 $*$
HawkesKT [44]	0.6815 ± 0.0041 $*$	0.6905 ± 0.0025 $*$
qDKT [36]	0.6986 ± 0.0006 $*$	0.6922 ± 0.0005 $*$
QIKT [4]	0.7260 ± 0.0013 $*$	0.7077 ± 0.0014 $*$
IEKT [22]	0.7301 ± 0.0012 $*$	0.7106 ± 0.0018 $\circ$
LPKT [35]	<u>0.7340</u> ± 0.0007 $\circ$	<u>0.7128</u> ± 0.0004 $\circ$
	Our 0.7355 ± 0.0006	**0.7134** ± 0.0005

features of question q_t can only attend to tokens corresponding to $q_{\leq t}$ and $r_{<t}$. In combination with teacher forcing, these changes enable us to efficiently train a knowledge tracing model based on our student representation.

We experiment on large-scale student data provided with Ednet [6], which includes learning data from 784,309 students. The *pykt*-benchmark [21] enables standardized comparison against several recent KT models. In Table 3 we provide an evaluation of our model against the best performing baselines.[8] The KT model based on our student representation achieves significantly better performance than all but one baselines and is on par with the remaining one. We conjecture that the excellent performance of our (adapted) response model can be taken as an indicator of the predictive accuracy of our response model in the CAT setting.

4.3 Uncertainty sampling

Proposition 1 suggests that uncertainty sampling provides a strong baseline given a good response model. We have observed good performance of uncertainty sampling in the CAT experiment in Section 4.1 and provided empirical evidence that our response model architecture is highly competitive on the CAT related knowledge tracing task in Section 4.2. We now investigate the interplay between response model, learned policy and uncertainty sampling for CAT tasks with uncalibrated questions in the question bank.

We generate artificial students as follows: Every student has a question bank of the same 50 candidate questions that are equally distributed in five groups. Within a group, a student either answers all questions correctly or incorrectly. This setting resembles questions belonging to knowledge components that are either fully understood or completely unknown to the student. Whether a student

[8] We report on the eleven baselines achieving the best AUC out of 22 total comparisons.

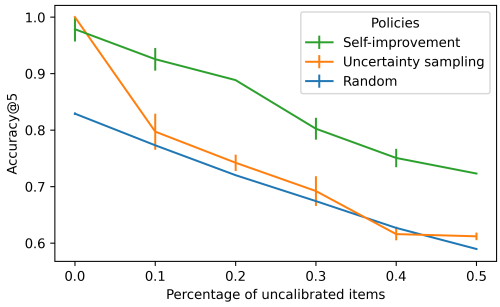

Fig. 3: Performance of different policies on artificially generated CAT tasks with differing proportions of uncalibrated items; error bars indicate standard error.

has mastered a group is determined randomly by a coin flip. In addition, we include *uncalibrated* questions that are answered randomly according to a coin flip as well, but that are not linked to any group and for which the policy cannot acquire any information. We experiment with different percentages of these uncalibrated questions in the question bank. We train on 500 students and validate and test on 1000 students. We report on averages over five repetitions.

Figure 3 shows the results for test length $T = 5$. As expected, uncertainty sampling achieves perfect results in the absence of uncalibrated questions, by picking one question of every group each, thereby reducing the uncertainty of the response model towards that group. However, as soon as we observe uncalibrated questions, uncertainty sampling deteriorates and self-improvement dominates in terms of predictive accuracy.

4.4 Discussion

We conclude that our student representation and response model lead to accurate predictions and is well suited for question selection algorithms in CAT-based learning tasks. However, under optimal conditions, a simple uncertainty sampling based policy is sufficient to achieve perfect results, as we have shown theoretically in Proposition 1 and experimentally on artificial data. Hence, our response model together with uncertainty sampling should be sufficient for adaptive testing in ideal settings. If, however, calibration tests result in a suboptimal question bank due to non-trivial student guessing and slip probabilities, a learned policy can improve predictive accuracies. With self-improvement training to jointly learn deep response and policy models, we provide a strong solution to CAT under suboptimal conditions.

5 Related Work

Early on, computerized adaptive testing (CAT) allowed for large-scale personalized tests [43] and is closely related to item response theory (IRT, [23,29]).

While CAT can reduce test lengths by assessing students with fewer items and in shorter time due to an item selection algorithm based on previous responses, IRT provides the foundation of explicitly modeling latent abilities of students. Recent approaches to policy learning for CAT include reinforcement learning; for example [10] propose shallow feed-forward neural networks as their policy model and [51] employ an attention based neural network architecture, where optimization is based on the score function estimator [47] and deep Q-learning [25], respectively. Formulating CAT as a subset selection problem, [52] propose a greedy algorithm based on the approximation of expected gradient differences. Response models are either based on IRT, simple feed-forward neural networks or on a neural cognitive diagnosis model (CDM) as proposed in [45] and account for explicit modeling of latent student skills. For a more comprehensive survey on modern CAT approaches see [19].

Without an explicit focus on interpretability, our student representation and response model are more closely related to recent deep learning approaches to knowledge tracing (KT, [7]), a related educational field which is concerned with reasoning about changes in students' knowledge and enabling the adaptation of learning materials. Transformer-based KT models include for example [27,5,9,20]. Closely related to our response model is [50], who also consider an alternating sequence of question and response embedding as model input. Different from our approach, their sequential model includes question features (such as knowledge components) only indirectly via optimization of auxiliary prediction tasks.

Besides training with self-improvement [28], the design of our policy model is further related to recent work in the field of neural combinatorial optimization. Specifically, [41] introduced classification based on cross-attention scores for sequential combinatorial problems where the target dictionary size depends on the input at the current time step. The idea has since successfully been applied to policy learning for combinatorial optimization via reinforcement learning [16]. Our policy architecture is closely related to the traveling salesman problem (TSP) transformer [2], which provides a similar architecture to the one proposed in this paper. However, we operate on sets of sets of tokens rather than only a set of tokens like the TSP transformer and we allow for different features in encoder and decoder, corresponding to questions and question-response tuples, respectively.

6 Conclusion

We studied computerized adaptive testing (CAT) as an iterative subset selection problem, learning both the underlying response model for cognitive diagnosis and a policy for question selection jointly from offline training data (e.g., a calibration pre-study). We leveraged the close relation of CAT to neural combinatorial optimization (NCO), proposed novel response and policy models, and adapted a recent self-improvement training approach to CAT policy learning, relying on strong generalization properties of deep learning models.

Our proposed response model empirically outperformed baselines in CAT as well as in related knowledge tracing tasks. We further provided theoretical and

empirical evidence that our response model can be combined successfully with uncertainty sampling-based policies in scenarios where the response model can be learned (almost) perfectly. Our results also show that scenarios with imperfect response models (e.g., due to higher guess and slip probabilities) clearly favor jointly learning both a response and policy model via self-improvement training as proposed in this paper.

Avenues for future work include (i) exploiting the novel student representation by introducing more descriptive features to CAT, (ii) exploring more sampling schemes [17,28] for self-improvement training in CAT, and (iii) combining our approach with interpretable models or post-hoc interpretability methods [8,31].

Acknowledgments. We thank Laurin Luttmann for discussions on advances in NCO. We thank the Joachim Herz Foundation for their support and funding as part of the ALEE project. Infrastructure used in this project was funded in parts by the European Union (EFRE/85202549).

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bresson, X., Laurent, T.: The transformer network for the traveling salesman problem. arXiv preprint arXiv:2103.03012 (2021)
3. Chang, H.H., Ying, Z.: A global information approach to computerized adaptive testing. Applied Psychological Measurement **20**(3), 213–229 (1996)
4. Chen, J., Liu, Z., Huang, S., Liu, Q., Luo, W.: Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 14196–14204 (2023)
5. Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C., Heo, J.: Towards an appropriate query, key, and value computation for knowledge tracing. In: Proceedings of the seventh ACM conference on learning @ scale (2020)
6. Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., Heo, J.: Ednet: A large-scale hierarchical dataset in education. In: Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21. pp. 69–73. Springer (2020)
7. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**, 253–278 (1994)
8. Gervet, T., Koedinger, K., Schneider, J., Mitchell, T., et al.: When is deep learning the best approach to knowledge tracing? Journal of Educational Data Mining **12**(3), 31–54 (2020)
9. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2330–2339 (2020)
10. Ghosh, A., Lan, A.: Bobcat: Bilevel optimization-based computerized adaptive testing. In: International Joint Conference on Artificial Intelligence (2021)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. Huang, J., Gu, S.S., Hou, L., Wu, Y., Wang, X., Yu, H., Han, J.: Large language models can self-improve. In: Conference on Empirical Methods in Natural Language Processing (2023)
13. Im, Y., Choi, E., Kook, H., Lee, J.: Forgetting-aware linear bias for attentive knowledge tracing. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3958–3962 (2023)
14. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: International Conference on Learning Representations (2014)
15. Kitaev, N., Kaiser, Ł., Levskaya, A.: Reformer: The efficient transformer. In: International Conference on Learning Representations (2020)
16. Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! In: International Conference on Learning Representations (2019)
17. Kool, W., Van Hoof, H., Welling, M.: Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In: International Conference on Machine Learning. pp. 3499–3508. PMLR (2019)
18. Lewis, D.D.: A sequential algorithm for training text classifiers: Corrigendum and additional data. In: Acm Sigir Forum. vol. 29, pp. 13–19 (1995)
19. Liu, Q., Zhuang, Y., Bi, H., Huang, Z., Huang, W., Li, J., Yu, J., Liu, Z., Hu, Z., Hong, Y., et al.: Survey of computerized adaptive testing: A machine learning perspective. arXiv preprint arXiv:2404.00712 (2024)
20. Liu, Z., Liu, Q., Chen, J., Huang, S., Luo, W.: simpleKT: a simple but tough-to-beat baseline for knowledge tracing. In: International Conference on Learning Representations (2023)
21. Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., Luo, W.: pyKT: a python library to benchmark deep learning based knowledge tracing models. In: Advances in Neural Information Processing Systems. vol. 35, pp. 18542–18555 (2022)
22. Long, T., Liu, Y., Shen, J., Zhang, W., Yu, Y.: Tracing knowledge state with individual cognition and acquisition estimation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 173–182 (2021)
23. Lord, F.: A theory of test scores. Psychometric monographs (1952)
24. Lord, F.M.: Applications of item response theory to practical testing problems. Routledge (2012)
25. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
26. Novick, M.R.: The axioms and principal results of classical test theory. Journal of Mathematical Psychology **3(1)**, 1–18 (1966)
27. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. In: 12th International Conference on Educational Data Mining, EDM 2019. pp. 384–389. International Educational Data Mining Society (2019)
28. Pirnay, J., Grimm, D.G.: Self-improvement for neural combinatorial optimization: Sample without replacement, but improvement. Transactions on Machine Learning Research (2024)
29. Rasch, G.: Probabilistic models for some intelligence and attainment tests. Danmarks Paedagogiske Institut (1960)
30. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In: International Conference on Machine Learning. pp. 1278–1286 (2014)

31. Rodrigues, T.B., de Souza, J.F., Bernardino, H.S., Baker, R.S.: Towards interpretability of attention-based knowledge tracing models. In: Anais do XXXIII Simpósio Brasileiro de Informática na Educação. pp. 810–821. SBC (2022)
32. Rothe, S., Narayan, S., Severyn, A.: Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics **8**, 264–280 (2020)
33. Settles, B., T. LaFlair, G., Hagiwara, M.: Machine learning–driven language assessment. Transactions of the Association for computational Linguistics **8**, 247–263 (2020)
34. Shen, S., Huang, Z., Liu, Q., Su, Y., Wang, S., Chen, E.: Assessing student's dynamic knowledge state by exploring the question difficulty effect. In: Proceedings of the 45th international ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 427–437 (2022)
35. Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., Wang, S.: Learning process-consistent knowledge tracing. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1452–1460 (2021)
36. Sonkar, S., Waters, A.E., Lan, A.S., Grimaldi, P.J., Baraniuk, R.G.: qDKT: Question-Centric Deep Knowledge Tracing. In: 13th International Conference on Educational Data Mining (2020)
37. Spearman, C.: 'General intelligence,' objectively determined and measured. The American Journal of Psychology **15(2)**, 201–293 (1904)
38. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
39. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
41. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer Networks. In: Advances in Neural Information Processing Systems. pp. 2692–2700 (2015)
42. Wainer, H., Dorans, N.J., Flaugher, R., Green, B.F., Mislevy, R.J.: Computerized adaptive testing: A primer. Routledge (2000)
43. Wainer, H., Kiely, G.L.: Item clusters and computerized adaptive testing: A case for testlets. Journal of Educational measurement **24**(3), 185–201 (1987)
44. Wang, C., Ma, W., Zhang, M., Lv, C., Wan, F., Lin, H., Tang, T., Liu, Y., Ma, S.: Temporal cross-effects in knowledge tracing. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 517–525 (2021)
45. Wang, F., Liu, Q., Chen, E., Huang, Z., Chen, Y., Yin, Y., Huang, Z., Wang, S.: Neural cognitive diagnosis for intelligent education systems. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6153–6161 (2020)
46. Wang, Z., Lamb, A., Saveliev, E., Cameron, P., Zaykov, J., Hernandez-Lobato, J.M., Turner, R.E., Baraniuk, R.G., Barton, C., Jones, S.P., et al.: Results and Insights from Diagnostic Questions: The NeurIPS 2020 Education Challenge. In: NeurIPS 2020 Competition and Demonstration Track. pp. 191–205. PMLR (2021)
47. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**, 229–256 (1992)
48. Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., Chen, E., Li, X.: Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023. pp. 855–864 (2023)

49. Zhang, C., Bütepage, J., Kjellström, H., Mandt, S.: Advances in Variational Inference. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 2008–2026 (2019)
50. Zhou, H., Rong, W., Zhang, J., Sun, Q., Ouyang, Y., Xiong, Z.: AAKT: Enhancing Knowledge Tracing with Alternate Autoregressive Modeling. IEEE Transactions on Learning Technologies (2024)
51. Zhuang, Y., Liu, Q., Huang, Z., Li, Z., Shen, S., Ma, H.: Fully adaptive framework: Neural computerized adaptive testing for online education. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 4734–4742 (2022)
52. Zhuang, Y., Liu, Q., Zhao, G., Huang, Z., Huang, W., Pardos, Z., Chen, E., Wu, J., Li, X.: A bounded ability estimation for computerized adaptive testing. In: Advances in Neural Information Processing Systems. vol. 36, pp. 2381–2402 (2023)