No Task Left Behind: Multi-Task Learning of Knowledge Tracing and Option Tracing for Better Student Assessment

Suyeong An*, Junghoon Kim*, Minsam Kim*, Juneyoung Park†

Riiid AI Research, Seoul, Republic of Korea
{suyeong.an, junghoon.kim, minsam.kim, juneyoung.park}@riiid.co

Abstract

Student assessment is one of the most fundamental tasks in the field of AI Education (AIEd). One of the most common approach to student assessment is Knowledge Tracing (KT), which evaluates a student's knowledge state by predicting whether the student will answer a given question correctly or not. However, in the context of multiple choice (polytomous) questions, conventional KT approaches are limited in that they only consider the binary (dichotomous) correctness label (i.e., correct or incorrect), and disregard the specific option chosen by the student. Meanwhile, Option Tracing (OT) attempts to model a student by predicting which option they will choose for a given question, but overlooks the correctness information. In this paper, we propose Dichotomous-Polytomous Multi-Task Learning (DP-MTL), a multi-task learning framework that combines KT and OT for more precise student assessment. In particular, we show that the KT objective acts as a regularization term for OT in the DP-MTL framework, and propose an appropriate architecture for applying our method on top of existing deep learning-based KT models. We experimentally confirm that DP-MTL significantly improves both KT and OT performances, and also benefits downstream tasks such as Score Prediction (SP).

Introduction

The field of AI Education (AIEd) is concerned with developing AI systems that facilitate human learning, and has the potential to provide personalized education to a wider audience at an affordable cost. Student assessment, the process of evaluating a student's knowledge level, is one of the most fundamental tasks in AIEd. Proper student assessment can then be used for many downstream educational tasks, such as score prediction (SP) (Su et al. 2018; Yin et al. 2019; Choi et al. 2021) and personalized content recommendation (Chen, Lee, and Chen 2005; Wang 2008; Wang et al. 2016; Ai et al. 2019).

Knowledge Tracing (KT) (Corbett and Anderson 1994) models a student's knowledge state by predicting whether the student will answer a given question correctly or not. Due to the method's simplicity and domain-agnosticity, KT has been extensively used for student assessment in AIEd (Choi et al. 2020a; Piech et al. 2015). However, for educational contents that involve multiple choice (polytomous) questions, KT only considers the binary (dichotomous) label of correctness, and does not consider the student's option of choice. KT thus fails to distinguish between students who answered a given question incorrectly, when in reality, one student's answer might have been closer to the correct one than the other student's.

Option Tracing (OT) (Ghosh, Raspat, and Lan 2021; Thissen and Steinberg 1984) is an approach that explicitly models the student's response, i.e., option choice, to a multiple choice question. While OT takes into account the student's option choice information, it does not consider the student's correctness. As a result, OT may fail to trace the student's knowledge state properly. This motivates a multi-task learning scheme that leverages both correctness labels and option labels.

In this paper, we propose Dichotomous-Polytomous Multi-Task Learning (DP-MTL), where the model learns to predict both the student's correctness and option choice for a given question. This way, DP-MTL can track the student's knowledge state at a more granular level. In our experiments, we demonstrate that DP-MTL indeed improves KT, OT, and SP performances. We expect that DP-MTL will enable a more accurate student representation learning, which would, in turn, benefit many other downstream educational tasks.

The main contributions of our paper are as follows:

- We introduce DP-MTL and show that, in this framework, the KT objective acts as a regularization term for OT.
- Additionally, we propose an architecture design for combining KT and OT on top of existing KT models.
- We experimentally confirm that our method significantly improves KT, OT, and SP performances when applied on top of three popular KT models, based on two different datasets.

To the best of our knowledge, this is the first work that combines KT and OT for better student assessment.

*These authors contributed equally.
†This author is the corresponding author.

Related Works

Knowledge Tracing

Knowledge Tracing (KT) is a student assessment task that models a student's knowledge state by predicting whether a student will answer a given question correctly or not. Dichotomous Item Response Theory (D-IRT) models (Kingston and Dorans 1982; Way and Reese 1990; Chen, Lee, and Chen 2005) predict the student's answer correctness using extracted user and item parameters, where a user parameter and an item parameter each corresponds to the user's skill and the question's difficulty, respectively.

Collaborative filtering (CF) method, which is equivalent to the multi-dimensional D-IRT method (except for the absence of sigmoid function) (Vie and Kashima 2019), models each user (item) as a user (item) vector, instead of a scalar value. Though CF was originally developed for recommendation systems (Koren, Bell, and Volinsky 2009), CF has been extensively used for KT (Khosravi, Cooper, and Kitto 2017; Vie and Kashima 2019). Recently, Neural Matrix Factorization (NMF) methods proposed to replace the conventional dot product used in CF with neural network computations (He et al. 2017; Xue et al. 2017).

Sequential KT approaches model the student's learning trajectory, as opposed to modeling the interactions at an atomic level as done in D-IRT and CF. Bayesian Knowledge Tracing (BKT) (Corbett and Anderson 1994) is the original KT method, which traces the student's knowledge state based on hidden Markov model. Recently, a lot of research effort went into applying various deep learning architectures for sequential KT, including RNN-based models (Piech et al. 2015; Minn 2020), Dynamic Key-Value Memory Networks (DKVMN) (Zhang et al. 2017), and transformer-based models (Pandey and Karypis 2019; Choi et al. 2020a).

All the aforementioned methods consider KT exclusively, and do not perform OT. We apply our proposed method DP-MTL on top of three popular KT methods, namely, (1) D-IRT, (2) collaborative filtering, and (3) LSTM-based KT, and demonstrate that DP-MTL consistently improves the KT performance across all models considered.

Option Tracing

Option Tracing (OT) (Ghosh, Raspat, and Lan 2021) is a student assessment task that traces a student's knowledge state by predicting the student's exact answer choice given a multiple choice question. Polytomous IRT (P-IRT), in an analogous manner to D-IRT, predicts the student's option choice using the extracted user parameters and item option parameters. Recently, Ghosh, Raspat, and Lan (2021) proposed to perform OT based on modified deep KT models for a more accurate student assessment. However, (1) they did not consider a multi-task learning setup that simultaneously performs both KT and OT; and (2) their architecture cannot take account of subtle details when performing OT (e.g., permuting the options (A,B,C) to (B,A,C) will not change the prediction from (p_A, p_B, p_C) to (p_B, p_A, p_C)). We not only consider the multi-task learning of KT and OT, but also propose an appropriate deep learning architecture accordingly.

Score Prediction

Student score prediction (SP) is another important student assessment task we consider in this work (Sweeney, Lester, and Rangwala 2015; Iqbal et al. 2017; Loh, Chae, and Hwang 2020). Prior SP methods rely on collaborative filtering (Elbadrawy and Karypis 2016; Sweeney et al. 2016), and regression models (Morsy and Karypis 2017; Ren et al. 2019). Other recent methods utilize KT algorithms, and address SP as a downstream task (Liu et al. 2019; Choi et al. 2021). We also treat SP as a downstream task, and show that DP-MTL leads to improved SP performance.

Multi-Task Learning

Multi-Task Learning (MTL) (Caruana 1997) is a machine learning approach that trains a model to perform several related tasks simultaneously. MTL posits that training signals from a particular domain help form inductive bias for other related tasks. In this work, we demonstrate that KT and OT are an example of such related tasks that, when performed simultaneously, are mutually beneficial.

Methodology

In this section, we propose Dichotomous-Polytomous Multi-Task Learning (DP-MTL) that learns to perform both KT and OT simultaneously. In particular, we show that in the DP-MTL framework, the multi-task learning objective has an explicable interpretation: the KT loss acts as a regularization term for OT. Also, we propose an architecture design necessary for applying DP-MTL on top of existing KT models.

Notations

This section introduces the notations that will be used throughout this paper.

Users Within the total population of n students, each student u, $1 \leq u \leq n$, has a d dimensional user parameter $\boldsymbol{\theta}_u \in \mathbb{R}^d$. Each dimension should model a user's skill level in a particular knowledge component (KC).

Choices/Options The possible choice set for each question i with j total multiple choice options is denoted by $\mathcal{O}_i = \{o_i^1, o_i^2, \ldots, o_i^j\}$. In other words, user u's choice for given item i is $o_{u,i} \in \mathcal{O}_i$. The correct choice o_i^* for given item(question) i must always be within the set of choices ($o_i^* \in \mathcal{O}_i$).

Items For a standardized multiple choice question examination with a total of m questions, each question i ($1 \leq i \leq m$) with choice k ($1 \leq k \leq j$) has d dimensional item parameters per each choice, $\mathbf{a}_{i,k} = (a_{i,k}^1, a_{i,k}^2, \ldots, a_{i,k}^d) \in \mathbb{R}^d$. For simplicity, we denote $\mathbf{a}_i$ to be the set of item parameters with each choice $\mathbf{a}_i = (\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \ldots, \mathbf{a}_{i,j})$

DP-MTL Training

Dichotomous Option Correctness (D) The conventional dichotomous model is trained by minimizing the negative log likelihood of observing the interactions that consists of the user, item, and the pair's corresponding *correctness*. This

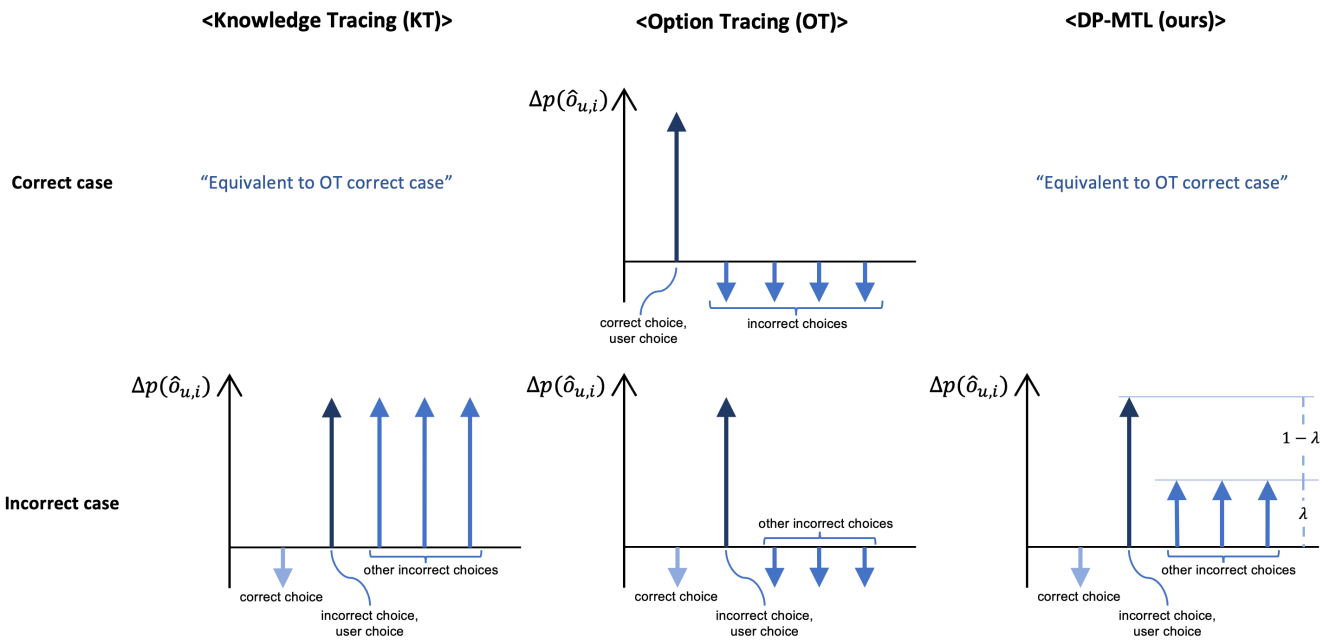

Figure 1: Illustration of DP-MTL. When the student chooses the correct answer, KT, OT, and DP-MTL are equivalent (Upper). However, when the student chooses the incorrect answer, DP-MTL interpolates between KT and OT. (Lower).

is equivalent to maximizing the conditional probability of the user responding correctly/incorrectly to the item, based on the student interaction data. In other words,

$$L_D(y_{u,i};\boldsymbol{\theta}_u,\mathbf{a}_i) = y_{u,i}\log P(\hat{y}_{u,i}=1|\theta_u,\mathbf{a}_i) + (1-y_{u,i})\log P(\hat{y}_{u,i}=0|\boldsymbol{\theta}_u,\mathbf{a}_i) \quad (1)$$

is minimized, where $\hat{y}_{u,i}$ is the prediction for u getting the question i correctly, and $y_{u,i}$ is the correctness label included in $\{0,1\}$.

Polytomous Option Choice (P) Training of a polytomous model is done by minimizing the negative log likelihood $L_P(\theta_u,\mathbf{a}_i)$ of a user u responding to a question i with choice $o_{u,i}$:

$$L_P(o_{u,i};\boldsymbol{\theta}_u,\mathbf{a}_i) = \log P(\hat{o}_{u,i}=o_{u,i}|\boldsymbol{\theta}_u,\mathbf{a}_i). \quad (2)$$

Here, $\hat{o}_{u,i}$ is the predicted categorical variable that represents the option choice of student u for the given question i, and $o_{u,i}$ is the option label.

Substituting $P(\hat{y}_{u,i}=0|\boldsymbol{\theta}_u,\mathbf{a}_i)$ from Equation 1 with the sum of probabilities of incorrect choices $\sum_{o_i^j\neq o_i^*} P(\hat{o}_{u,i}=o_i^j|\boldsymbol{\theta}_u,\mathbf{a}_i)$, L_D becomes

$$L_D = y_{u,i}\log P(\hat{o}_{u,i}=o_i^*|\boldsymbol{\theta}_u,\mathbf{a}_i) + (1-y_{u,i})\log[\sum_{o_i^j\neq o_i^*} P(\hat{o}_{u,i}=o_i^j|\boldsymbol{\theta}_u,\mathbf{a}_i)] \quad (3)$$

DP-MTL DP-Multi Task Learning (DP-MTL) is a combined version of option correctness (D) and option choice (P) with a ratio of $\lambda : 1-\lambda$ where $0 \leq \lambda \leq 1$. Abbreviating notations for simplicity's sake, we define DP-MTL's training objective as follows:

$$L_{DP} = \lambda L_D + (1-\lambda)L_P \quad (4)$$

The objective function of DP-MTL could be thus derived by simply combining the two objective functions.

That is, for all u,i such that $o_{u,i} = o_i^*$, given the user answer was correct,

$$L_{DP}(y_{u,i},o_{u,i};\boldsymbol{\theta}_u,\mathbf{a}_i) = \log P(\hat{o}_{u,i}=o_i^*|\boldsymbol{\theta}_u,\mathbf{a}_i), \quad (5)$$

and for all u,i such that $o_{i,j} \neq o_i^*$, given the user answer was incorrect we have:

$$L_{DP}(y_{u,i},o_{u,i};\boldsymbol{\theta}_u,\mathbf{a}_i) = \lambda\log[\sum_{o_i^j\neq o_i^*} P(\hat{o}_{u,i}=o_i^j|\boldsymbol{\theta}_u,\mathbf{a}_i)] + (1-\lambda)\log P(\hat{o}_{u,i}=o_{u,i}|\boldsymbol{\theta}_u,\mathbf{a}_i). \quad (6)$$

Note that the objective function is equivalent to that of Equation 1 when the user chooses the correct option o_i^*, regardless of λ value, as shown in Equation 5. However, if the user answers incorrectly $o_{u,i} \neq o_i^*$ for a given item, not only does the likelihood that the user selects the specific choice increase, but also does the likelihood corresponding to the other incorrect options, proportional to λ. Figure 1 provides a schematic of how DP-MTL controls likelihood for each option.

Although the underlying motivation is a simple weighted average between the two tasks' losses within a multi-task learning framework, the resulting objective Equation 6 shows that the two tasks connect intuitively in a form of regularization. From an option tracing perspective, increasing

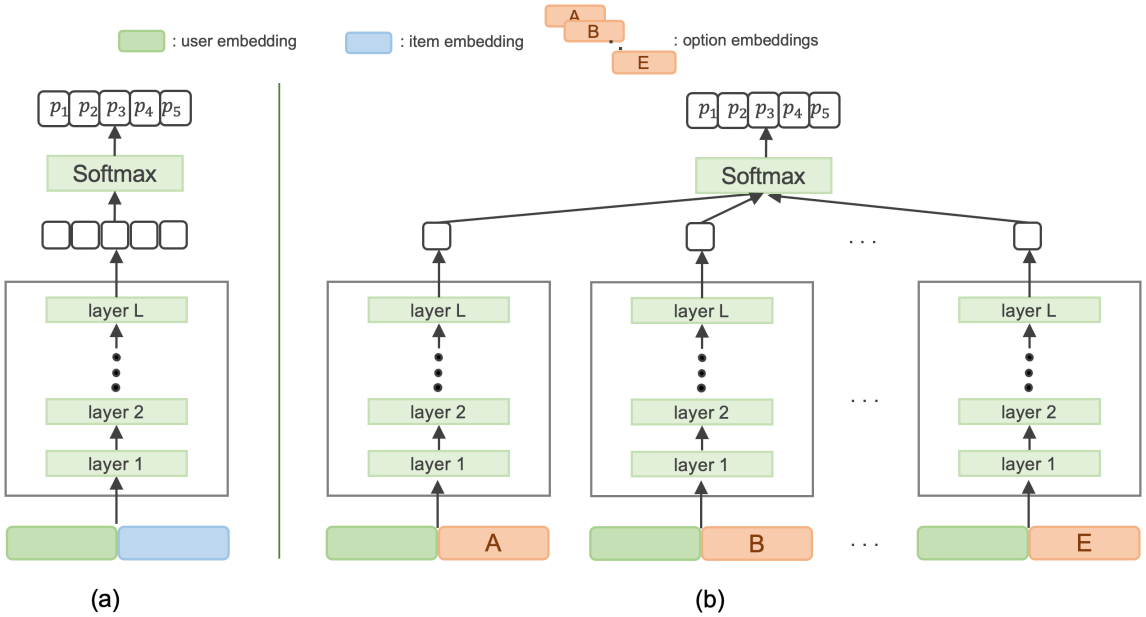

Figure 2: (a) Previous Option Tracing does not use any option information and relies on pre-defined positions for each choice. (b) Our proposed architecture instead uses option embeddings directly to output logits for each option, independent of positions.

λ from 0 to 1 gradually limits the model's discrimination among the incorrect options.

As an example where such regularization might help, consider an exam consisting of questions with incorrect option choices that do not discriminate students' skill level to any significant degree (i.e. an exam with low discrimination between student cohorts). Under such circumstances, the optimal λ value will be large, and thus the DP-MTL model will be reduced down to a dichotomous option correctness model.

Applying DP-MTL to KT Models

In this subsection, we explain how to apply our proposed DP-MTL framework to existing KT models.

DP-IRT is the application of DP-MTL to a CF-based option tracing model extended from Vie and Kashima (2019). Using Equation 4 as the objective function, we can combine KT and OT in a straightforward manner.

DP-NMF is the application of DP-MTL to NMF (He et al. 2017) with some tweaks, as applying DP-MTL to NMF is less trivial than the case of the above DP-IRT. Recently, Ghosh, Raspat, and Lan (2021) proposed to perform OT with NMF. However, they treated OT as a multi-class classification problem without giving option information as the input. For instance, given a set of 5-choice problems, to generate the predictions they share a 5-class Softmax layer as in Figure 2. This approach may encounter two issues:

- Positional Bias - Prone to learning noises that we do not want to model. For example, information like "a user may habitually guess for option C" or "the real answer was often option B" can be modeled.
- Functionality - Unable to handle cases when the number of choices differ for each question, or when the multiple choice order of each question is mixed for different users.

To address these issues, in DP-NMF, we provide option information to the input and generate separate output representations for each option.

DP-BiDKT is the application of DP-MTL to DKT (Piech et al. 2015) with similar tweaks based on the same reasoning as in the above DP-NMF (Figure 2). Note that for datasets we consider, we use a Bidirectional LSTM (Hochreiter and Schmidhuber 1997), hence the name BiDKT. We would also like to point out that we can easily apply similar modifications to Transformer-based (Vaswani et al. 2017) models (Pandey and Karypis 2019; Choi et al. 2020a).

Experiment Setup

The proposed DP-MTL framework is evaluated on the three models explained in Section 3 (DP-IRT, DP-NMF, DP-BiDKT) and three tasks (KT, OT, SP), based on two different datasets (ENEM, TOEIC). KT and OT serve as primary tasks, while Score Prediction (SP) serves as a downstream task, where we evaluate the quality of the student representation obtained from KT and OT. For KT, we measure the performance with ROC-AUC (Area under the ROC) as it is a binary classification task, while for OT, we use Accuracy as performance measure. For SP, a regression task of predicting the student's exam score, we use Mean Absolute Error (MAE) as performance measure.

In our experiments, we first obtain student and question representations via DP-MTL framework, then fit a simple score prediction model based on training user/student split. The score prediction module consists of user representation

Dataset	Num. Users	Num. Questions	Sparisty	Type	Description
ENEM	10k	185	0%	Exam	Brazil's standardized national college entrance exam.
TOEIC	9877	13399	4%	Snapshot	User's 2-week snapshot from Choi et al. (2020b)

Table 1: A summary of the two datasets used in our experiments: ENEM and TOEIC.

Model	Hyper-parameter	Search Space
Common	Mixing Ratio λ	$\{0.0, 0.1, ...1.0\}$
	Embedding Dimension	$\{1, 4, 8, 16, 32, 64\}$
NMF Bi-DKT	Num. Layers	$\{1, 2, 3, 4\}$

Table 2: The table shows the hyper-parameter search space for all models and datasets considered in our experiments.

fed into a simple linear regression model followed by isotonic regression. The model's performance is measured by the test MAE metric, which also serves as a quality measure for the student representation θ from DP-MTL framework.

Datasets

ENEM ENEM dataset in our experiment consists of 10000 students' question solving record on 185 questions from 2019 Exame Nacional do Ensino Medio (ENEM) examination. In ENEM, every student solved all 185 questions, thus providing a dense matrix of students and questions. For score prediction task label, we use sum of the 4 section scores for each student[1].

TOEIC 9877 active users' interaction dataset within an online Intelligent Tutoring System for preparing Test of English for International Communication (TOEIC) exam was used as an additional real-life dataset. The students solved different sets of questions within a question bank of 13399 questions. The score dataset consists of the students' self-reported official TOEIC score out of a total score of 990[2].

Sparsity Ablation

Due to the large number of questions in TOEIC, the original dataset EdNet (Choi et al. 2020b) is extremely sparse. For inference performance and sparsity ablation, three different versions of the original dataset is created. Only top N% of questions (columns) solved by most students and N% of students (rows) who solved most questions are preserved, where N is set to be 10, 25, and 50. Thus, Top 10% version yields smallest and most dense (4% sparsity) interaction matrix, while Top 50% version yields largest and most sparse (74% sparsity) one.

Hence, to verify our methodology's robustness against situations with data sparsity, different versions of ENEM training interaction dataset were also created by randomly dropping the student-question interaction pair at different ratios (0%, 10%, ..., 70%). In the following section, we report all three tasks' results based on all imposed sparsity ratios.

[1]The entire code, ENEM, and TOEIC datasets are available at: https://github.com/godtn0/DP-MTL

[2]Due to privacy issues, we do not release score prediction dataset for TOEIC.

Results and Discussion

In order to evaluate the performance of the proposed DP-MTL framework, two sets of evaluation results are reported. First, the performance difference based on different values of λ is reported to demonstrate the value of multi-task learning in creating a holistic student representation. Second, the individual performances of individual models on various sparsity levels is reported to determine the most effective MTL model for the tasks of KT and SP.

Impact of DP-MTL: λ Ablation

Since different tasks and datasets yield significantly different scales of performance metrics, configuration-wise performance **rank** of eleven λ values (0.0, 0.1, ..., 1.0) were averaged across different datasets and models. The result is shown in Figure 3. Each line corresponds to different tasks of SP, KT, and OT. Smaller y-axis value of average rank indicates that the performance is relatively superior. For instance, using λ value of either 0 or 1 performs significantly worse than λ values closer to 0.5, consistently for all three tasks. This convexity serves as a strong empirical evidence of our proposed DP-MTL framework's advantage over the two extreme baseline approaches of KT and OT. We highlight that the multi-task learning of task A (KT) and B (OT) not only improved metrics on the down-stream task C (SP), but also improved the metrics of the original tasks A and B.

Score Prediction All three individual models separately show the desired convex shape of rank average metric with respect to λ parameter, as shown in Figure 4. The degree of improvement from KT and OT baselines is largest in DP-NMF model, which has relatively larger number of trainable parameters than the other two models.

We also note that different models show different trend of optimal λ with respect to data sparsity. For DP-NMF, most λ hyper-parameters are chosen to be 0.6 and 0.7, consistently. Figure 5 shows the heatmap of SP-MAE metrics from ENEM dataset standardized within each sparsity ratio setup. Large continuous blue region of smaller MAE emphasizes the advantage from introducing λ persists across stable range and across different data sparsity ratios. As opposed to DP-NMF, DP-BiDKT's optimal λ value gradually decreases as the dataset sparsity increases. The proposed DP-MTL framework allows the model to tune its attention between KT and OT.

Knowledge Tracing and Option Tracing From KT-AUC block of Table 3, most optimal λ values in ENEM dataset are

		SP-MAE			KT-AUC		
Dataset	**Sparsity**	DP-BiDKT	DP-IRT	DP-NMF	DP-BiDKT	DP-IRT	DP-NMF
ENEM	0%	**43.1(0.8)**	48.2(0.9)	50.2(0.7)	0.7373(0.0)	0.7356(0.9)	**0.7381(0.1)**
	10%	**52.1(0.8)**	60.4(0.7)	62.3(0.6)	**0.7374(0.0)**	0.73(0.3)	0.7346(0.4)
	20%	**56.6(0.5)**	58.6(0.6)	60.9(0.4)	**0.7363(0.5)**	0.7324(0.3)	0.7345(0.2)
	30%	**60.1(0.3)**	67.0(0.6)	68.6(0.6)	**0.7291(0.1)**	0.7233(0.1)	0.7272(0.3)
	40%	**71.9(0.4)**	75.1(0.4)	77.2(0.6)	0.7213(0.1)	**0.7228(0.1)**	0.7178(0.3)
	50%	**83.5(0.3)**	85.6(0.5)	93.5(0.7)	0.7084(0.0)	**0.7113(0.5)**	0.7035(0.3)
	60%	**104.0(0.1)**	142.0(0.0)	115.5(0.7)	**0.6957(0.0)**	0.6647(0.0)	0.6934(0.9)
	70%	**125.5(0.0)**	166.4(0.8)	197.9(0.6)	**0.637(0.1)**	0.6192(0.9)	0.6161(0.3)
TOEIC_Top10	4%	**62.2(1.0)**	76.7(1.0)	72.9(1.0)	**0.7699(0.1)**	0.7661(0.9)	0.7481(0.6)
TOEIC_Top25	47%	**58.2(0.9)**	69.2(1.0)	69.5(1.0)	0.7809(0.6)	**0.7826(0.9)**	0.774(0.9)
TOEIC_Top50	74%	**59.1(0.6)**	69.8(1.0)	69.5(0.6)	**0.849(0.6)**	0.7961(0.8)	0.7864(0.9)

Table 3: The table represents the test SP-MAE and KT-AUC for each dataset-model configuration. The figures in the brackets represent the best λ value.

		OT-ACC		
Dataset	Sparsity	DP-BiDKT	DP-IRT	DP-NMF
ENEM	0%	**0.3851(0.1)**	0.3842(0.6)	0.3818(0.0)
	10%	**0.389(0.4)**	0.3831(0.2)	0.3843(0.3)
	20%	**0.3854(0.4)**	0.3846(0.2)	0.3827(0.2)
	30%	**0.3824(0.1)**	0.3815(0.1)	0.3789(0.5)
	40%	0.3769(0.6)	**0.3801(0.4)**	0.375(0.3)
	50%	0.3636(0.5)	**0.3656(0.7)**	0.3534(0.0)
	60%	**0.3455(0.2)**	0.3422(0.4)	0.3392(0.0)
	70%	0.3004(0.3)	**0.3131(0.0)**	0.279(0.4)
TOEIC10	4%	**0.6746(0.3)**	0.6679(0.0)	0.6563(0.2)
TOEIC25	47%	**0.6992(0.5)**	0.6973(0.1)	0.6912(0.2)
TOEIC50	74%	**0.7421(0.6)**	0.7144(0.2)	0.7142(0.2)

Table 4: The table represents the test OT-ACC for each dataset-model configuration. The figures in the brackets represent the best λ value.

closer to 0, as opposed to 1. In other words, tackling OT task alone yielded better results in terms of KT-AUC for ENEM dataset. This trend is particularly strong in TOEIC dataset, as shown in Figure 6. For all three models, focusing on KT alone (rank 10) yields worse KT-AUC performance than focusing on OT alone (rank 8). Furthermore, λ value between 0.6 and 0.9 leads to sharp improvement of performance(rank 2-4).

Model Comparison

Based on the hyper-parameter configuration chosen from validation set performance, test performance metrics for ENEM and TOEIC dataset's are provided in Table 3. First block represents results on Score Prediction-Mean Absolute Error (SP-MAE), and the second block represents Knowledge Tracing-Area Under ROC Curve (KT-AUC). The model entry with best performance is highlighted in bold for each task, and the figures in brackets represent the chosen λ parameter in our DP-MTL framework. We reiterate that $\lambda = 1$ corresponds to KT/D-IRT scenario, while $\lambda = 0$ corresponds to OT/P-IRT scenario.

Score Prediction We compare the three models in SP task where the assessment is focused on the quality of the extracted student representation. DP-BiDKT significantly outperforms the other models by large margin, consistently across different datasets of different sparsity ratios. Under most sparse conditions, SP-MAE reduction is as high as **15.3%** and **24.9%** for ENEM and TOEIC dataset, respectively. In general, DP-NMF's capability of fitting into nonlinear patterns beyond DP-IRT is not providing any advantage in the SP task.

Knowledge Tracing and Option Tracing Although DP-BiDKT model's outperformance is not as significant as that in score prediction task, the model achieved top results in most settings in both knowledge tracing and option tracing task. (Option tracing result is shown in the Appendix Table 4.) Also, in the most sparse TOEIC_Top50 dataset, improvement of DP-BiDKT model over the first-runner-up in both of KT and OT task metrics are **6.6%** and **3.9%**.

In summary, the empirical results strongly support the efficacy of the proposed DP-MTL framework on all three tasks assessing the quality of user-item representation. The multi-task learning approach for KT and OT not only yielded optimal for the down-stream SP task, but also for KT and OT themselves. Furthermore, our DP-BiDKT architecture achieved significant improvement over standard baseline algorithms by efficient parameter reduction/reusing and novel encoding of user interaction sequence.

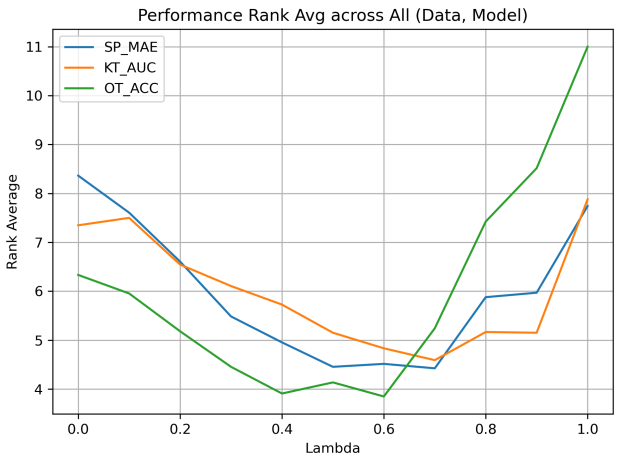

Figure 3: Impact of DP-MTL's λ parameter on KT, OT, and SP - y-axis denotes averaged performance rank across all datasets and models in each task. x-axis denotes λ. Performance rank is a convex curve on the space of λ. regardless of metrics.

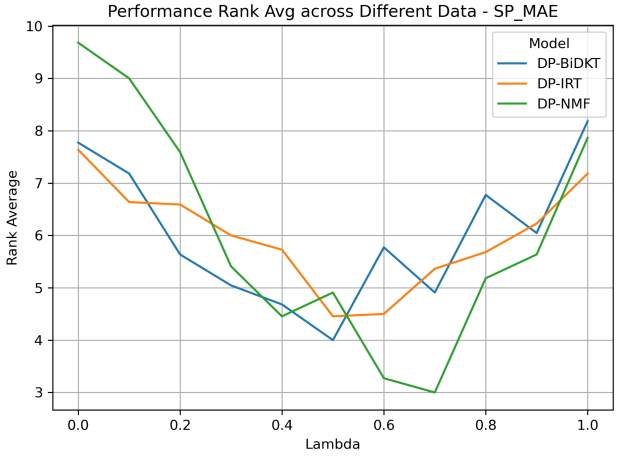

Figure 4: Impact of DP-MTL's λ parameter on SP-MAE - y-axis denotes averaged performance rank across all of datasets in SP. x-axis denotes λ.

Conclusion

This study proposed a multi-task learning framework to include (a) response correctness and (b) the specific response choice of a student to provide a more holistic student assessment model that outperforms the existing single-task baselines. Extensive empirical results from the two datasets and the three tasks (1) showed significant improvement upon existing models (IRT, CF, NMF) and (2) revealed intriguing relationship between KT and OT under various data sparsity conditions. In addition, customized DP-BiDKT architecture was proposed to further improve parameter efficiency and simplify input encoding under our DP-MTL framework, which yielded best performance in most experiment settings.

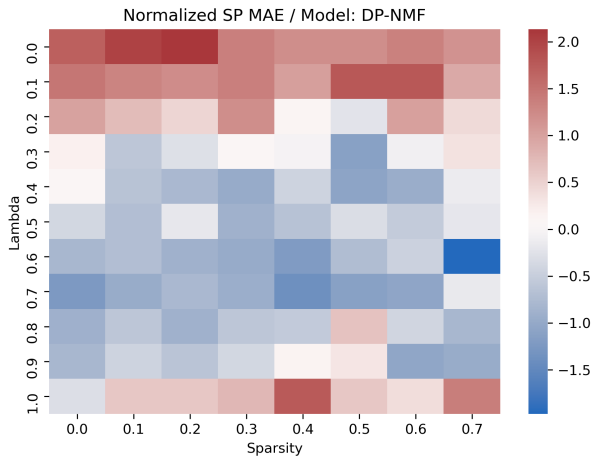

Figure 5: Normalized SP-MAE with DP-NMF in ENEM - Each cell denotes the averaged SP-MAE across all dimensions in conditions with λ and sparsity.

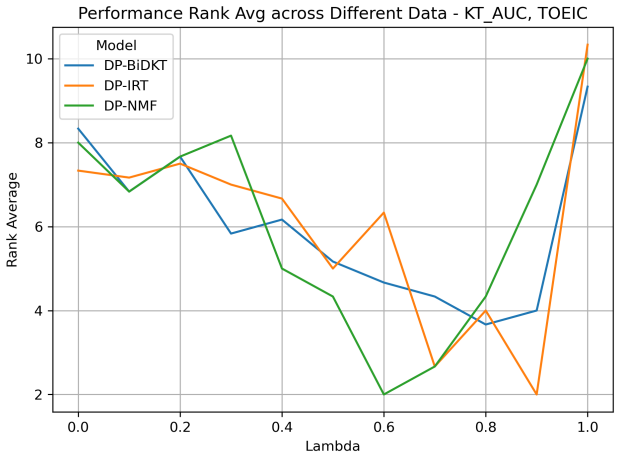

Figure 6: KT-AUC vs λ, TOEIC

Beyond improving KT/OT performance, this work provides an example where better user-item representation can benefit separate down-stream tasks such as student score prediction. Other potential future applications include individualized educational content recommendation and weakness identification based on improved representation learning of students and educational contents.

References

Ai, F.; Chen, Y.; Guo, Y.; Zhao, Y.; Wang, Z.; Fu, G.; and Wang, G. 2019. Concept-Aware Deep Knowledge Tracing and Exercise Recommendation in an Online Learning System. *International Educational Data Mining Society*.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chen, C.-M.; Lee, H.-M.; and Chen, Y.-H. 2005. Personal-

ized e-learning system using item response theory. *Computers & Education*, 44(3): 237–255.

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

Choi, Y.; Lee, Y.; Cho, J.; Baek, J.; Shin, D.; Yu, H.; Shim, Y.; Lee, S.; Shin, J.; Bae, C.; Kim, B.; and Heo, J. 2021. Assessment Modeling: Fundamental Pre-training Tasks for Interactive Educational Systems. arXiv:2002.05505.

Choi, Y.; Lee, Y.; Shin, D.; Cho, J.; Park, S.; Lee, S.; Baek, J.; Bae, C.; Kim, B.; and Heo, J. 2020b. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, 69–73. Springer.

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

Elbadrawy, A.; and Karypis, G. 2016. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 183–190.

Ghosh, A.; Raspat, J.; and Lan, A. 2021. Option Tracing: Beyond Correctness Analysis in Knowledge Tracing. In *International Conference on Artificial Intelligence in Education*, 137–149. Springer.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Iqbal, Z.; Qadir, J.; Mian, A. N.; and Kamiran, F. 2017. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*.

Khosravi, H.; Cooper, K.; and Kitto, K. 2017. RiPLE: Recommendation in peer-learning environments based on knowledge gaps and interests. *arXiv preprint arXiv:1704.00556*.

Kingston, N. M.; and Dorans, N. J. 1982. The feasibility of using item response theory as a psychometric model for the GRE Aptitude Test. *ETS Research Report Series*, 1982(1): i–148.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

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

Loh, H.; Chae, P.; and Hwang, C. 2020. Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons. In *EDM*.

Minn, S. 2020. BKT-LSTM: Efficient Student Modeling for knowledge tracing and student performance prediction. *arXiv preprint arXiv:2012.12218*.

Morsy, S.; and Karypis, G. 2017. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 552–560. SIAM.

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

Piech, C.; Spencer, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. *arXiv preprint arXiv:1506.05908*.

Ren, Z.; Ning, X.; Lan, A. S.; and Rangwala, H. 2019. Grade Prediction Based on Cumulative Knowledge and Co-taken Courses. *International Educational Data Mining Society*.

Su, Y.; Liu, Q.; Liu, Q.; Huang, Z.; Yin, Y.; Chen, E.; Ding, C.; Wei, S.; and Hu, G. 2018. Exercise-enhanced sequential modeling for student performance prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Sweeney, M.; Lester, J.; and Rangwala, H. 2015. Next-term student grade prediction. In *2015 IEEE International Conference on Big Data (Big Data)*, 970–975. IEEE.

Sweeney, M.; Rangwala, H.; Lester, J.; and Johri, A. 2016. Next-term student performance prediction: A recommender systems approach. *arXiv preprint arXiv:1604.01840*.

Thissen, D.; and Steinberg, L. 1984. A response model for multiple choice items. *Psychometrika*, 49(4): 501–519.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vie, J.-J.; and Kashima, H. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 750–757.

Wang, F.-H. 2008. Content recommendation based on education-contextualized browsing events for web-based personalized learning. *Journal of Educational Technology & Society*, 11(4): 94–112.

Wang, Z.; Zhu, J.; Li, X.; Hu, Z.; and Zhang, M. 2016. Structured knowledge tracing models for student assessment on coursera. In *Proceedings of the third (2016) ACM conference on learning@ scale*, 209–212.

Way, W. D.; and Reese, C. M. 1990. An investigation of the use of simplified IRT models for scaling and equating the TOEFL test. *ETS Research Report Series*, 1990(2): i–22.

Xue, H.-J.; Dai, X.; Zhang, J.; Huang, S.; and Chen, J. 2017. Deep Matrix Factorization Models for Recommender Systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 3203–3209.

Yin, Y.; Liu, Q.; Huang, Z.; Chen, E.; Tong, W.; Wang, S.; and Su, Y. 2019. Quesnet: A unified representation for heterogeneous test questions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1328–1336.

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