The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Neighborhood-Aware Negative Sampling for Student Knowledge and Behavior Modeling

Siqian Zhao, Sherry Sahebi

Department of Computer Science, University at Albany - SUNY
szhao2@albany.edu, ssahebi@albany.edu

Abstract

Simple random negative sampling is a technique used to enhance decision-making in sequential models with numerous potential negative instances, like recommender systems. However, it ignores the patterns that can be discovered in complex sequences to select the most informative negative samples. In this paper, we address this challenge by introducing a Neighborhood-Aware Negative Sampling (NANS) technique in the context of student knowledge modeling (KM) and behavior modeling (BM). In the education domain, KM quantifies student knowledge based on past performance, while BM focuses on behaviors like student preferences of questions. With the vast number of problems to choose from and the intricate relationship between student knowledge and behavior, selecting the proper negative samples becomes a notable challenge in this problem. NANS, along with our proposed multi-objective, multi-task sequential model for KM and BM, NANS-KoBeM frames the simultaneous modeling of student knowledge and question selection as a multi-task learning problem with dual objectives: predicting students' performance and their question selections.

1 Introduction

Negative sampling is used in decision-making tasks with a large sample space to select a small subset of potentially negative samples to contrast with one or a few observed positive ones. Traditionally, a simple uniform random negative sampling technique is employed, where items that users have not interacted with are randomly selected as negative samples at each time step for sequential decision-making (Chae et al. 2018; Zhang et al. 2013; Rendle and Freudenthaler 2014). However, even with this technique, predicting the next sequence item is difficult, as uniform sampling does not distinguish informative versus non-informative samples. Recently, more advanced negative sampling techniques, such as distribution-aware sampling (Gillick et al. 2019; Xiong et al. 2020; Liu et al. 2023), have been proposed to alleviate these problems. However, these techniques mostly ignore the signals of more complex sequences with rich patterns and interrelations to select the most informative negative samples.

In this work, we present Neighborhood-Aware Negative Sampling (NANS) to select the most informative negative samples for complex sequences, using their coarse-grained patterns, to help predict the fine-grained ones. NANS builds an item graph to discover the shared coarse-grained sequential similarities for weeding out the potentially trivial and non-informative samples and selecting a more representative negative set, thereby improving both training efficiency and prediction accuracy.

We propose NANS in the context of student knowledge modeling (KM) and behavior modeling (BM) tasks that present complex interrelations and sequential patterns. KM aims to quantify student knowledge to predict their future performance, and BM focuses on student behavior, such as predicting the questions they choose to work with. Consider an online education system, where numerous questions are presented, and the students can choose any topic and question to attempt at a time. Specifically, while the students are free to choose the questions in any order, research has shown that student sequences follow a coarse-grained pattern driven by latent topic structures and question similarities (Zhao, Wang, and Sahebi 2020). On the other hand, fine-grained analyses of student sequences show that the specific question they choose to attempt at each step is more personalized and depends on both student knowledge and behavioral traits (Guerra et al. 2014; Mirzaei 2020). These results indicate that such sequences share coarse-grained similarities with fine-grained differences: the consecutive questions are usually related, with a high variance in the order they are tried. As a result, predicting the next question a student chooses to attempt can significantly benefit from NANS.

Furthermore, the above research results highlight the importance of modeling the interrelations between KM and BM, which is under-investigated in the literature. To model these interrelations, we further propose KoBeM, a deep sequential model that includes two interconnected MANN-based components: one focused on student knowledge acquisition and the other on learning student preferences, with mechanisms for information transfer between them. We formulate this problem as a multi-objective multi-task learning challenge, to predict: (1) the next question that the student chooses to engage with, and (2) the student's performance in it. Finally, we apply Pareto Multi-Task Learning (MTL) optimization to balance the two objectives of KM and BM.

Our extensive experiments on two real-world datasets indicate the significant improvement of NANS-KoBeM over

---

the baseline methods in both student performance prediction and next question prediction tasks, the effectiveness of NANS compared to alternative negative sampling methods, in both efficiency and effectiveness, and the importance of modeling interrelations between student knowledge and behavior modeling. Our main contributions in this paper are summarized as follows:

- We propose a novel neighborhood-aware negative sampling method, NANS, that is the first negative sampling technique introduced in the context of student knowledge and behavior modeling and is empowered by a coarse-grain item similarity graph to select the most informative samples at each time;
- We propose an innovative multi-objective multi-task student knowledge and behavior model, KoBeM, which emphasizes the importance of modeling the interrelations between these tasks, enables the information transfer between them, and balances their objectives; and
- We conduct extensive experiments on NANS-KoBeM, comparing NANS with alternative negative sampling methods, KoBeM with baseline KM and BM models, and ablation studies on KoBeM, demonstrating significant improvements and highlighting the importance of each proposed component.

2 Related Work

2.1 Knowledge Modeling and Behavior Modeling

KM, also known as knowledge tracing, aims to estimate a student's knowledge state at each learning step as they interact with a question (Thai-Nghe, Horváth, and Schmidt-Thieme 2010; Sahebi, Lin, and Brusilovsky 2016).Recent advancements have introduced recurrent neural networks (RNNs) to enhance the modeling of knowledge states (Piech et al. 2015; Zhao, Wang, and Sahebi 2022). Additionally, memory-augmented neural networks (MANNs) have been developed to more accurately capture knowledge changes (Zhao and Sahebi 2023; Wang, Zhao, and Sahebi 2021; Zhang et al. 2017). Attention mechanisms have also been employed to address interdependencies and long-term dependencies, further improving the understanding of student learning patterns (Pandey and Karypis 2019; Ghosh, Heffernan, and Lan 2020).

BM seeks to understand student behavior patterns (Mirzaei 2020; VanLehn 1988), such as learning habits (Chrysafiadi and Virvou 2013), preferences for learning materials (Zhao and Sahebi 2024a,c,b), and engagement levels (Goldberg et al. 2021). Research has examined aspects such as gamified learning, dropout rates, retention, and participation frequency (Mirzaei 2020; Zahedi 2019; Kumar, Singh, and Handa 2017; Zahedi et al. 2020; Dalipi, Imran, and Kastrati 2018; Tornwall, Lu, and Xie 2020).

However, the approaches mentioned above have focused solely on either KM or BM, overlooking the relationship between student behavior and knowledge. Recent studies suggest that modeling both simultaneously can enhance understanding (Zhao and Sahebi 2024c,a,b), but they typically focus on learning material types. In contrast, our work targets student behavior in selecting specific questions, which poses a large sample space challenge.

2.2 Negative Sampling

Negative sampling was introduced to accelerate skip-gram training in word2vec by sampling negative examples based on word frequency (Mikolov et al. 2013; Yang et al. 2020). It has also been widely applied in recommender systems (RS), initially using uniform sampling to improve recommendation performance (Rendle et al. 2012). Subsequently, graph-based negative sampling techniques were developed for collaborative filtering (CF) (Huang et al. 2021; Yang et al. 2022) and graph contrastive learning (Xia et al. 2021; Zhu et al. 2022, 2021; Yang et al. 2020). In RS, MixGCF (Yang et al. 2022) generates synthetic negatives by aggregating embeddings from different layers of raw negatives' neighborhoods. RecNS (Huang et al. 2021) introduces the three-region principle, suggesting more negatives be sampled from intermediate regions while minimizing sampling in adjacent and distant regions. GNNO (Fan et al. 2023) constructs an item transition graph and uses Jaccard similarity to measure neighborhood overlap, generating negative samples from items with medium or zero similarity to the target item. In graph contrastive learning, STENCIL (Zhu et al. 2022) utilizes global topology features for negative sampling, and ProGCL (Xia et al. 2021) addresses false negative discrimination by using a two-component beta mixture model to distinguish true negatives. Additionally, MCNS (Yang et al. 2020) samples negative items from the entire item set in graph representation learning, assuming the edge distribution of negatives is sub-linearly related to that of positives.

In our work, we focus on sampling the most informative negative items by selecting those most similar to the target item based on coarse-grained similarities in an item neighborhood graph. Applied in the context of student behavior with questions as items, this approach leverages coarse and fine-grained patterns in student sequences, creating a more challenging training task that enhances accuracy and generalizability. Additionally, sampling complexity during inference is significantly reduced as samples are drawn from a pre-established neighborhood set.

3 Problem Formulation

Consider an online learning system with M students and a set of questions $\mathbb{Q}$, where each student chooses one question at a time to practice and receive feedback on their performance (e.g., score, correctness, or grade). Our goal is to jointly model both student preference behavior and their knowledge by predicting which question a student will choose next and how they will perform on it. Let (q_t, r_t) represent a student's activity at a given time step t, where $q_t \in \mathbb{Q}$ is the question the student interacts with, and $r_t \in \{0,1\}$ is the student's performance on that question. The sequence of a student's historical learning activities is denoted as $\{(q_1, r_1), \ldots, (q_t, r_t)\}$. Our goals are to efficiently and effectively (1) predict the specific question q_{t+1} that the student will select at the next time step $t+1$, and (2) predict the student's performance r_{t+1} on that selected question q_{t+1}, given the student's historical learning activities. As the large question label space ($Q = |\mathbb{Q}|$), poses challenges to both efficiency and effectiveness of our goals, our third goal is to choose the best negative sample set of questions ($\mathbb{Q}^-_{q_t} \in \mathbb{Q}$) at each training step to alleviate these challenges.

4 NANS-KoBeM

Our framework, NANS-KoBeM, includes the proposed Neighborhood-Aware Negative Sampling (NANS) technique and multi-objective, multi-task sequential model (KoBeM). NANS-KoBeM's architecture is presented in Figure 2. It consists of four main components: (1) a neighborhood-aware negative sampler that builds an item neighborhood graph and samples a set of negative items at each time step, (2) a multi-task knowledge and behavior modeling component with two MANN-based connected sub-components: one for knowledge tracing and another for preference modeling, (3) a prediction component for predicting the next question and student performance, and (4) a multi-objective model learning component that formulates two objectives and employs Pareto Multi-Task Learning (MTL) optimization to balance them.

4.1 Neighborhood-Aware Negative Sampling (NANS)

NANS aims to use the coarse-grained patterns presented in item sequences to select the most informative negative samples for comparison against the observed positive sample. As distinguishing the most similar items from the target item is more challenging for classifiers, such items could be more informative, as they could help the model train faster, be more generalizable, and improve the overall prediction performance. Moreover, considering that consecutively interacted items demonstrate latent item similarities, NANS builds a coarse-grained transition graph and samples the negative items guided by this graph. While we propose NANS in the student knowledge and behavior modeling context, it can be applied to any sequential decision-making task with coarse-grained patterns like implicit consecutive items similarities. In the following, we detail the transition graph and sampling strategy of NANS.

Weighted Question Transition Graph NANS facilitates the discovery of coarse-grain item similarities by creating a graph from the sequence of students' learning activities in the training set. For this, we construct a graph $\mathcal{G} = (\mathbb{Q}, \mathcal{E}, \mathcal{W})$, where $\mathbb{Q}$ consists of all questions as nodes, and $\mathcal{E}$ represents the undirected edges between questions, corresponding to transitions between questions in student sequences. An edge exists between two questions if a student from the training sessions has interacted with them consecutively. Each edge between nodes q_i and q_j has a weight $w_{q_i,q_j} = \frac{|I_{q_i,q_j}|}{\sum_{i,j}|I_{q_i,q_j}|}$, where $|I_{q_i,q_j}|$ denotes the number of times q_i and q_j have been interacted with consecutively in all training students' sequences, and $\sum_{i,j}|I_{q_i,q_j}|$ totals all transitions from one question to another in all training student learning activity sequences. The edge weight represents the similarity between the two questions. A toy example is illustrated in Figure 1. In this example, there are a total of nine transitions between questions. Specifically, there are two transitions between q_1 and q_5, q_2 and q_4, and q_2 and q_5. Thus, the edge weight between q_1 and q_2, q_2 and q_4, and q_2 and q_5 are all $\frac{2}{9}$. The weights for all other edges are $\frac{1}{9}$.

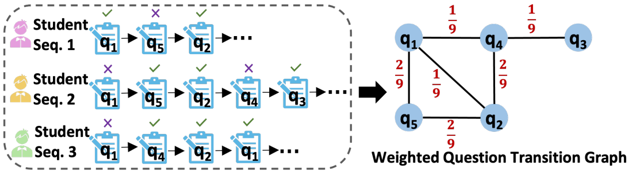

Figure 1: An illustration of constructing a weighted question transition graph from a toy example of three sequences.

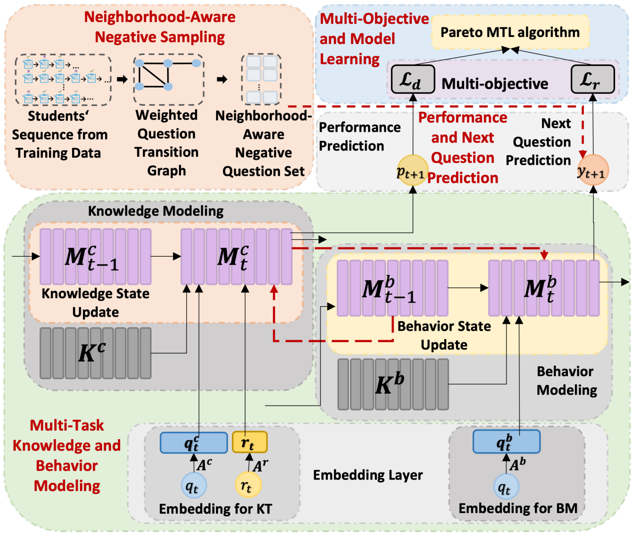

Figure 2: NANS-KoBeM architecture.

Neighborhood-Aware Negative Question Set To select the most informative negative samples, we first construct a set of neighboring questions that many students have interacted with consecutively. Then, for each question q_t, we select a set of N (a hyperparameter) nearest neighbors as its candidate negative sample set $\mathbb{Q}^-_{q_t}$ based on the weights of its edges ($w_{q_t,*}$). These neighbors are the nodes with the highest weight values $w_{q_t,*}$. We then sample the next negative samples from this candidate sample set to predict q_{t+1}. Note that the weight is only used to build the candidate sample set, and the final negative items are sampled uniformly from this set. This practice further improves the generalizability. For example, if we consider selecting the two nearest neighbors as the negative sample set for a question, in the toy example in Figure 1, the negative sample set for q_2 would consist of q_4 and q_5, excluding the other neighbor,q_1.

4.2 Multi-Task Knowledge and Behavior Modeling

To simultaneously model the relationship between KM and BM, we propose two interconnected components: one focused on student knowledge acquisition for KM and the other on learning student preferences for BM, with mechanisms for information transfer between these components.

We first map each student activity (q_t, r_t) to obtain high-dimensional activity representations as the input and use two variants of Memory-Augmented Neural Networks (MANNs) to model student knowledge and behavior preferences. Two static key matrices, $\boldsymbol{K}^b \in \mathbb{R}^{n_b \times d_b}$ and $\boldsymbol{K}^c \in \mathbb{R}^{n_c \times d_c}$, store n_b and n_c latent behavioral features and knowledge concepts, characterized by d_b and d_c dimensions, respectively. Additionally, we incorporate two dynamic value matrices, $\boldsymbol{M}_t^b \in \mathbb{R}^{n_b \times v_b}$ and $\boldsymbol{M}_t^c \in \mathbb{R}^{n_c \times v_c}$, to concurrently update and store behavioral preferences and student knowledge at each time step, with v_b and v_c representing the memory slot sizes. The interaction between student knowledge and behavior is captured through information transfer facilitated by these dynamic memory matrices.

Embedding Layer Given that the latent features of student knowledge and behavior may differ, we use separate question embedding matrices for KM and BM to embed each question in student activities, capturing distinct information for modeling knowledge and behavior. First, in BM, questions are mapped into the latent behavioral preference feature space by using the embedding matrix $\boldsymbol{A}^b \in \mathbb{R}^{Q \times d_b}$, where d_b is the embedding size of the latent features. Similarly, We retrieve $\boldsymbol{A}^c \in \mathbb{R}^{Q \times d_c}$ as questions' latent concepts embeddings matrix for KM, where d_q is the embedding size of latent concepts. Moreover, we utilize another matrix $\boldsymbol{A}^r \in \mathbb{R}^{2 \times d_r}$ to embed binary student performance r_t (success or failure), where d_r is the embedding size.

Correlation Weight To capture the student knowledge and behavior states, we first compute the correlation weight vectors $\boldsymbol{w}_t^b$ $\boldsymbol{w}_t^c$, representing the correlation between the question and each of the n_b latent behavioral features and n_c latent knowledge concepts. These weights are derived using an attention mechanism applied to the question embeddings $\boldsymbol{q}_t^b$ and $\boldsymbol{q}_t^c$, along with the key matrices $\boldsymbol{K}^b$ and $\boldsymbol{K}^c$:

$$w_t^b(i) = Softmax(\boldsymbol{q}_t^{b\top} \boldsymbol{K}^b(i)) \tag{1}$$

$$w_t^c(i) = Softmax(\boldsymbol{q}_t^{c\top} \boldsymbol{K}^c(i)) \tag{2}$$

where $w_t^b(i) \in \mathbb{R}^{n_b}$ and $w_t^c(i) \in \mathbb{R}^{n_c}$ represent the i-th elements of the correlation weight vectors $\boldsymbol{w}_t^b$ and $\boldsymbol{w}_t^c$, respectively. A $Softmax$ activation function is applied to ensure $\sum^{d_b} i = 1 w_t^b(i) = 1$ and$\sum^{d_c} i = 1 w_t^c(i) = 1$.

Behavior State Update We update the memory value matrix $\boldsymbol{M}_t^b$ to capture the student's question preferences at each time step. Since the knowledge state can influence behavioral preferences, we use the knowledge state $\boldsymbol{M}_t^v$ (Eq. 10) at time t as input to update the student preference behavior $\boldsymbol{M}_t^b$ for the same time step. Also, given the behavioral activity embedding $\boldsymbol{q}_t^b$ for question q_t at the current time step, we apply the *erase-and-add* mechanism to update $\boldsymbol{M}_t^b$, where redundant information from the last time step $\boldsymbol{M}_{t-1}^b$ is first erased using a vector $\boldsymbol{e}_t^b \in [0,1]^{v_b}$, followed by the addition of new information with a vector $\boldsymbol{a}_t^b \in [0,1]^{v_b}$, as follows:

Erase Step:

$$\boldsymbol{e}_t^b = Sigmoid(\boldsymbol{E}^{b\top} \boldsymbol{q}_t^b + \boldsymbol{E}_b^{c\top} \boldsymbol{M}_t^c + \boldsymbol{b}_e^b) \tag{3}$$

$$\tilde{\boldsymbol{M}}_t^b = \boldsymbol{M}_{t-1}^b(i) \cdot [\mathbf{1} - w_t^b(i) \boldsymbol{e}_t^b] \tag{4}$$

Add Step:

$$\boldsymbol{a}_t^b = Tanh(\boldsymbol{D}^{b\top} \boldsymbol{x}_t^b + \boldsymbol{D}_b^{c\top} \boldsymbol{M}_t^c + \boldsymbol{b}_d^b) \tag{5}$$

$$\boldsymbol{M}_t^b = \tilde{\boldsymbol{M}}_t^b(i) + w_t^b(i) \boldsymbol{a}_t^b \tag{6}$$

where $\oplus$ represents the concatenation operation, $\mathbf{1}$ is a vector with all values are 1, and $Sigmoid$ is the Sigmoid activation function. Matrices $\boldsymbol{E}^b, \boldsymbol{D}^b \in \mathbb{R}^{(d_b) \times v_b}$ are for mapping the question embedding of an activity to the behavior feature space. $\boldsymbol{E}_c^b$ and $\boldsymbol{D}_c^b \in \mathbb{R}^{v_c \times v_b}$ are for mapping the knowledge state to the behavior preference feature space. $\boldsymbol{b}_e^b$ and $\boldsymbol{b}_d^b \in \mathbb{R}^{d_v}$ are bias terms.

Knowledge State Update To trace student knowledge and model the impact of preference behavior on knowledge acquisition, the memory value matrix $\boldsymbol{M}_t^c$ is updated based on the embeddings of the student's activities at time t, capturing knowledge acquisition at that step, and the previous preference behavior state $\boldsymbol{M}_{t-1}^b$, reflecting the influence of behavior. The update also follows the mechanism as follows:

Erase Step:

$$\boldsymbol{e}_t^c = Sigmoid(\boldsymbol{E}^{c\top} [\boldsymbol{q}_t^c \oplus \boldsymbol{r}_t] + \boldsymbol{E}_b^{c\top} \boldsymbol{M}_{t-1}^b + \boldsymbol{b}_e^c) \tag{7}$$

$$\tilde{\boldsymbol{M}}_t^c = \boldsymbol{M}_{t-1}^c(i) \cdot [\mathbf{1} - w_t^c(i) \boldsymbol{e}_t^c] \tag{8}$$

Add Step:

$$\boldsymbol{a}_t^c = Tanh(\boldsymbol{D}^{c\top} [\boldsymbol{q}_t^c \oplus \boldsymbol{r}_t] + \boldsymbol{D}_b^{c\top} \boldsymbol{M}_{t-1}^b + \boldsymbol{b}_d^c) \tag{9}$$

$$\boldsymbol{M}_t^c = \tilde{\boldsymbol{M}}_t^c(i) + w_t^c(i) \boldsymbol{a}_t^c \tag{10}$$

where $\boldsymbol{E}^c, \boldsymbol{D}^c \in \mathbb{R}^{(d_c + d_r) \times v_c}$ are matrices for mapping the activity embedding to the concept feature space. $\boldsymbol{E}_b^c$ and $\boldsymbol{D}_b^c \in \mathbb{R}^{v_b \times v_c}$ are for mapping the preference behavior state to the concept feature space. $\boldsymbol{b}_e^c$ and $\boldsymbol{b}_d^c \in \mathbb{R}^{d_v}$ are bias terms.

4.3 Next Question and Performance Prediction

We have two prediction tasks: (1) to predict the student's upcoming question choice q_{t+1} and (2) their future performance r_{t+1} given the question q_{t+1}.

Next Question Prediction For predicting the student's next question choice, we first retrieve the read content $\boldsymbol{g}_t^b \in \mathbb{R}^{d_b + v_b}$ that reads and summarize the student behavioral preference from behavioral value matrices $\boldsymbol{M}_t^b$ as follows:

$$\boldsymbol{g}_t^b = \sum_{i=1}^{n_d} w_t(i) \boldsymbol{M}_t^b(i) \tag{11}$$

To estimate the probability of the next question the student will interact q_{t+1}, we concatenate the $\boldsymbol{g}_t^b$ with question q_t's behavioral embedding $\boldsymbol{q}_t$ and then pass it through a fully connected layer to obtain the summary vector $\boldsymbol{s}_t \in \mathbb{R}^{d_s}$ as a representation of the summary preference of student's next question where d_s is the dimension size of this converted summary vector. We then estimate the probability of the question the student will interact with as follows:

$$\boldsymbol{s}_t = Tanh(\boldsymbol{W}_s^\top [\boldsymbol{g}_t^b \oplus \boldsymbol{q}_t^b] + \boldsymbol{b}_s) \tag{12}$$

$$\boldsymbol{y}_{t+1} = sigmoid(\boldsymbol{W}_y^T \boldsymbol{s}_t + \boldsymbol{b}_y) \tag{13}$$

here, $Tanh$ is the Tanh activation function. The output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{Q}$ is a vector with a length equal to the number of questions, and each entry represents the predicted probability that the student will choose that question next. $\boldsymbol{W}_s \in \mathbb{R}^{(d_b+v_b)\times d_s}$ and $\boldsymbol{W}_y \in \mathbb{R}^{d_s \times Q}$ are weight matrices, $\boldsymbol{b}_s\mathbb{R}^{d_s}$ and $\boldsymbol{b}_y \in \mathbb{R}^{N}$ are bias terms.

Student Performance Prediction Given the question q_{t+1} that the student will choose to interact with at time step $t+1$, We first retrieve the read content $\boldsymbol{g}^{c}_{t+1} \in \mathbb{R}^{d_c+v_c}$, which summarizes the mastery level of knowledge for the question q_{t+1} from the knowledge memory value matrices $\boldsymbol{M}^{v}_{t}$ as follows:

$$\boldsymbol{g}^{c}_{t+1} = \sum_{i=1}^{n_c} w_{t+1}(i)\boldsymbol{M}^{c}_{t}(i) \tag{14}$$

where $w_{t+1}(i)$ is calculated using Eq. 2, with the embedding of the next question $\boldsymbol{q}_{t+1}$ used as the input. To predict the student performance of q_{t+1}, we then concatenate the read content of knowledge $\boldsymbol{g}^{c}_{t}$ and the question embedding $\boldsymbol{q}_{t+1}$ and then pass it through a fully connected layer to obtain the summary vector $\boldsymbol{f}_{t+1} \in \mathbb{R}^{d_f}$ where d_f is the dimension size of this summary vector. $\boldsymbol{f}_{t+1}$ contains both the student's mastery level and the question's concept information. We then calculate the student performance by using $\boldsymbol{f}_{t+1}$, as follows:

$$\boldsymbol{f}_{t+1} = Tanh(\boldsymbol{W}^{T}_{f}[\boldsymbol{g}^{c}_{t+1} \oplus \boldsymbol{q}^{c}_{t+1}] + \boldsymbol{b}_f) \tag{15}$$

$$\boldsymbol{p}_{t+1} = Sigmoid(\boldsymbol{W}^{T}_{p}\boldsymbol{f}_t + \boldsymbol{b}_p) \tag{16}$$

here, the output $\boldsymbol{p}_{t+1}$ is a scalar representing the predicted probability that the student will answer that particular question correctly. $\boldsymbol{W}_f \in \mathbb{R}^{(d_c+v_c)\times d_f}$ and $\boldsymbol{W}_p$ are weight matrices, $\boldsymbol{b}_s \in \mathbb{R}^{d_f}$ and $\boldsymbol{b}_p \in \mathbb{R}$ are bias terms.

4.4 Multi-Objective and Model Learning

For this multi-task learning challenge, where student knowledge modeling and preference behavior modeling are treated as interrelated tasks, we define two objectives.

Next Question Prediction Objective During the training process, for every question q_t at each time step and in every epoch, we randomly sample one negative question $\bar{q}_t$ from the neighborhood-aware negative sample set $\mathbb{Q}^{-}_{q_t}$. We retrieve the predicted probabilities from $\boldsymbol{y}_t$ in equation 13 for both the actual question q_t and the negative question $\bar{q}_t$, denoted as y^{p}_{t} and y^{n}_{t}, respectively. We label the positive actual interacted question and the negative sample question with z_t, where $z_t = 1$ for the actual sample and $z_t = 0$ for the negative sample. We compute the binary cross-entropy loss for these probabilities, using y^{p}_{t} and y^{n}_{t}, as follows:

$$\mathcal{L}_b = -\sum_{t}\sum_{\tilde{y}_t \in y^{p}_{t}, y^{n}_{t}} \left(z_t \log \tilde{y}_t + (1 - z_t)\log\left(1 - \tilde{y}_t\right)\right) \tag{17}$$

By doing so, we ensure that the predicted probability of the question a student actually interacted with is higher than that of the negative samples. This approach aims to better learn the student's behavioral preferences.

Student Performance Prediction Objective We employ the binary cross-entropy objective function to calculate the loss for student performance prediction at each time step t:

$$\mathcal{L}_r = -\sum_{t}\left(r_t \log p_t + (1 - r_t)\log(1 - p_t)\right) \tag{18}$$

where r_t represents the actual student performance, and p_t is the predicted student performance for the question q_t with which the student interacts.

Pareto Multi-Objective Optimization Combining $\mathcal{L}_r$ and $\mathcal{L}_b$ for dual objectives is difficult due to trade-off challenges (Lin et al. 2019). Multi-objective optimization strategies identify Pareto optimal solutions (Zitzler and Thiele 1999; Désidéri 2012). We use the Pareto MTL algorithm (Lin et al. 2019), which divides the problem into subproblems, generating balanced Pareto solutions for optimal prediction of student performance and the next question.

Time Complexity Building the weighted question transition graph is a one-time process based on the student activity history, with a time complexity of $O(|A|)$, where $|A|$ represents the total number of learning activities in the dataset. Sorting the neighbors of each question has a time complexity of $O(Q^2 \log Q)$. Each negative sampling retrieval operation has a basic cost of $O(1)$. Therefore, the overall time complexity for NANS is $O(|A| + Q^2 \log Q)$ with a constant retrieval time during inference. The time complexity of KoBeM for a student activity sequence of length L_s is $O\Big(L_s \cdot \big(n_c \cdot d_c^2 + n_c \cdot d_c + n_c \cdot v_c \cdot (d_c + d_r) + n_b \cdot d_b^2 + n_b \cdot d_b + n_b \cdot v_b \cdot d_b + v_c \cdot v_b\big)\Big)$.

5 Experiments

We evaluate the performance of NANS-KoBeM, comparing its predictive capabilities against baseline methods in tasks of student performance prediction and next-question type preference. These experiments include comparisons with baselines using alternative negative sampling strategies as well as ablations. Our codes are available at https://github.com/persai-lab/2025-NANSKoBeM.

5.1 Datasets

We use the two real-world datasets with general statistics presented in Table 1. **EdNet** (Choi et al. 2020b) is a publicly available anonymized dataset sourced from a multi-platform AI tutoring service, Santa[1], designed to help Korean students prepare for the TOEIC[2] English test. We use a preprocessed version of the dataset from previous studies (Zhao, Wang, and Sahebi 2022; Zhao and Sahebi 2023). **Junyi** (CMU DataShop 2015; Pojen, Mingen, and Tzuyang 2020) is another publicly available and anonymized dataset from the Chinese e-learning platform Junyi Academy[3], which is designed to teach children math. We use the preprocessed data introduced in (Chang, Hsu, and Chen 2015).

---

[1] https://www.aitutorsanta.com/

[2] https://www.ets.org/toeic

[3] https://official.junyiacademy.org/

Dataset	#Users	#Questions	Question Activities	Question Responses Mean	Question Responses STD	#Correct Question Responses	#Incorrect Question Responses
EdNet	1000	11249	200931	0.5910	0.2417	118747	82184
Junyi	2063	3760	290754	0.6660	0.2224	193664	97090

Table 1: Descriptive statistics of two datasets.

5.2 Baseline Methods

Next Question Prediction Baselines To evaluate the effectiveness of the NANS-KoBeM in predicting the next question, we first compare it to two *deep sequential* baseline models. **LSTM**(Hochreiter and Schmidhuber 1997) is known for its ability to capture long-term dependencies, the LSTM architecture is well-suited for tasks that require a deep understanding of complete data sequences. **MANN**(Santoro et al. 2016) enhances RNN with an external memory component, facilitating the storage and retrieval of information across extended sequences. This feature is particularly beneficial for tasks that necessitate sustained information retention and manipulation.

We also adopt other *negative sampling strategies* within our multi-task knowledge and behavior modeling framework to evaluate the effectiveness of NANS in modeling student behavior and improving the prediction of the next question. **RD-KoBeM** employs random negative sampling, where for each target question in a student's sequence, a completely random question from the entire question set is selected as the negative sample. **WD-KoBeM** utilizes weighted distribution negative sampling, where we construct a weighted question transition graph similar to NANS. However, instead of generating a negative sample from the neighborhood-aware negative question set, a negative question is generated based on the weight distribution between the target question and all other questions. **MCNS-KoBeM** leverages MCNS (Yang et al. 2020), a negative sampling strategy for graph representation learning, where negative items are sampled from the entire set of items. It assumes that the edge distribution of items being negative to the target items is sub-linearly related to the edge distribution of items being positive to the target items. **GNNO-KoBeM** employs GNNO (Fan et al. 2023), a negative sampling strategy originally proposed for the sequential recommendation. It constructs an item transition graph and uses Jaccard similarity to measure the degree of neighborhood overlap between the target item and others. The negative item is generated from a set of items with medium or zero similarity to the target item. Furthermore, we remove the KM component and eliminate the information transfer mechanism, retaining only the BM component, referred to as **NANS-KoBeM w/o KM**, to determine whether incorporating student knowledge modeling enhances the understanding of student behavior preferences.

Student Performance Prediction Baselines We evaluate the NANS-KoBeM's ability to model student knowledge and predict future performance by comparing it with six state-of-the-art supervised KM models: **DKT** (Piech et al. 2015): is the first deep learning-based KM model that uses RNN to model student knowledge acquisition. **DeepIRT** (Yeung 2019) is an extension of DKVMN that incorporates the one-parameter logistic item response theory (1PL-IRT) to reduce overfitting. **SAKT** (Pandey and Karypis 2019) employs a self-attentive model to capture relationships between activities at different time steps. **SAINT** (Choi et al. 2020a) is a transformer-based method that uses deep self-attentive layers to separately model questions and responses. **AKT** (Ghosh, Heffernan, and Lan 2020) is a context-aware model that uses a monotonic attention mechanism to aggregate past student performances relevant to the current question. **DKVMN** (Zhang et al. 2017) utilizes MANNs for KM, with a static key matrix for knowledge concepts and a dynamic value matrix for tracing student knowledge. It also named **NANS-KoBeM w/o BM**, is an ablation of our method that omits the modeling of student behavior and focuses solely on KM.

5.3 Experiment Setup

Evaluation Protocol In accordance with established evaluation protocols for sequential methods, as detailed in previous studies (Piech et al. 2015; Wang, Zhao, and Sahebi 2021; Zhao, Wang, and Sahebi 2022), we implement a 5-fold student-stratified cross-validation to divide the data and report the mean experiment results across five folds for each method and perform a paired t-test comparing each baseline to the NANS-KoBeM. For each fold, sequences from 80% of the students are designated as the training set, and sequences from the remaining 20% of the students are used as the testing set. Additionally, 20% of the training set is reserved for hyperparameter tuning. For the next question prediction task, we design an evaluation protocol inspired by the common evaluation protocol for sequential recommender systems, where the evaluation measures are computed out of a random selection of items instead of all potential items. Accordingly, for each question q_t at each time step t, we randomly sample 99 other questions (excluding q_t) as negative questions. We then evaluate the model's ability to rank the actual interacted question q_t against these negative questions. We use three widely used evaluation metrics for top-N recommendations: hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR). The top-N cutoff is set to $topN = 5$ for the generated rank list. For the student performance prediction task, we employ the Area Under the Curve (AUC) metric to assess the model's performance.

Implementation Details We use PyTorch[4] to develop the NANS-KoBeM. Consistent with established practices for handling sequential data (Lopez-del Rio et al. 2020; Dwarampudi and Reddy 2019), we standardize sequence lengths by either truncating or padding them with 0s as needed. The sequence length L_s is considered a hyperparameter. A coarse-grained grid search is performed across all methods, including baselines, to identify the optimal hyperparameters. The best hyperparameters of our NANS-KoBeM are reported in Table 2.

---

[4] https://pytorch.org/

Dataset	d_c	d_r	d_c	v_c	d_f	n_c	d_b	d_b	v_b	d_s	n_b	L_s	N
EdNet	64	32	32	32	32	32	32	32	32	32	16	50	10
Junyi	32	32	32	64	64	32	32	32	32	16	16	100	10

Table 2: Learned Best Hyperparameters of NANS-KoBeM

Methods	Ednet			Junyi		
	HR	NDCG	MRR	HR	NDCG	MRR
LSTM	0.0806^{**}	0.0452^{**}	0.0301^{**}	0.4156^{**}	0.2908^{**}	0.2611^{**}
MANN	0.0853^{**}	0.0515^{**}	0.0396^{**}	0.4284^{**}	0.3277^{**}	0.2838^{**}
RD-KoBeM	0.0973^{*}	0.0616^{*}	0.0498^{*}	0.4807^{*}	0.3709^{*}	0.3344^{**}
WD-KoBeM	0.0969^{*}	0.0614^{*}	0.0496^{*}	0.4355^{**}	0.3490^{**}	0.3200^{**}
MCNS-KoBeM	0.0972^{*}	0.0617^{*}	$\underline{0.0499^{*}}$	0.4871^{*}	0.3752^{*}	0.3387^{**}
GNNO-KoBeM	$\underline{0.0974^{*}}$	$\underline{0.0618^{*}}$	$\underline{0.0499^{*}}$	$\underline{0.4875^{*}}$	$\underline{0.3756^{*}}$	$\underline{0.3389^{**}}$
NANS-KoBeM w/o KM	0.0877^{**}	0.0555^{**}	0.0449^{**}	0.4391^{**}	0.3387^{**}	0.3051^{**}
NANS-KoBeM	**0.0982**	**0.0622**	**0.0503**	**0.4919**	**0.3795**	**0.3422**

Table 3: Next question prediction results. The best and second-best results are in boldface and underlined, respectively. $**$ and $*$ indicate paired t-test $p - value < 0.05$ and $p - value < 0.1$, compared to NANS-KoBeM.

5.4 Prediction Performance Comparison

The experimental results are presented in Tables 3 and 4 for next-question prediction and student performance.

Next Question Prediction Comparison NANS-KoBeM outperforms all baseline methods without a negative sampling strategy in the next-question prediction task across all metrics and datasets, underscoring its effectiveness in predicting the questions students will select. The superior performance of NANS-KoBeM on both datasets, compared to its ablation NANS-KoBeM w/o KM, which does not model the relationship between knowledge and behavior, demonstrates that simultaneously modeling students' knowledge and behaviors as a multi-task learning problem with multiple objectives significantly improves the understanding of their question preferences. Additionally, methods utilizing negative sampling consistently showed better results than those without it, indicating that the negative sampling strategy enhances the prediction of students' next question choices.

Alternative Negative Sampling Comparison We observed that NANS-KoBeM outperforms all baselines with negative sampling strategies across all metrics on both datasets. This demonstrates that neighborhood-aware negative sampling is more effective in the context of modeling student next-question selection behavior, enhancing the understanding of their question preferences. Moreover, GNNO-KoBeM and MCNS-KoBeM produce similar results to RD-KoBeM on both datasets, while WD-KoBeM performs worse on the Junyi dataset. This suggests that an unsuitable negative sampling method, such as GNNO and MCNS, may not offer more valuable insights into student preferences than random negative sampling. We observed that since each question has high-weighted neighbors, the generated negative samples from MD consistently tend to be those with higher weights. This lack of diversity in negative samples diminishes their effectiveness in capturing student preferences. These results further demonstrate that our NANS proves effective in the context of student next-question selection behavior, improving both training and prediction outcomes for next-question preferences.

Methods	EdNet	Junyi
	AUC	AUC
DKT	$\underline{0.6393^{**}}$	$\underline{0.8623^{**}}$
SAKT	0.6334^{**}	0.8053^{**}
SAINT	0.5205^{**}	0.7951^{**}
AKT	$\underline{0.6393^{**}}$	0.8093^{**}
DeepIRT	0.6290^{**}	0.8498^{**}
DKVMN (NANS-KoBeM w/o BM)	0.6296^{**}	0.8558^{**}
NANS-KoBeM	**0.6615**	**0.8779**

Table 4: Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. $**$ and $*$ indicate paired t-test $p - value < 0.05$ and $p - value < 0.1$, compared to NANS-KoBeM.

Student Performance Prediction Our experimental results demonstrate that NANS-KoBeM surpasses all baseline methods in predicting student performance across both datasets. This highlights NANS-KoBeM's ability to effectively track knowledge and accurately forecast student performance, emphasizing the importance of explicitly modeling student preferences for the next question in conjunction with knowledge modeling. When compared to NANS-KoBeM's ablation without modeling student behavior preferences, it is observed that NANS-KoBeM w/o BM performs worse than the full NANS-KoBeM model and, in some cases, even underperforms other KT baselines. These findings indicate that a multi-task learning model, which jointly models student behavior and knowledge while learning the associations between them, is highly effective. By representing these elements with distinct states and facilitating information transfer between them, NANS-KoBeM enhances our understanding of student knowledge and improves the accuracy of performance predictions.

6 Conclusions

In this paper, we addressed the problem of selecting informative negative samples for complex sequences by proposing Neighborhood-Aware Negative Sampling (NANS). NANS is the first negative sampling method introduced in the context of student knowledge and behavior modeling. We further proposed KoBeM, a multi-objective multi-task student knowledge and behavior model that can capture the intricate bidirectional relations between student knowledge and behavior while balancing their objectives. Our experiments showed that jointly modeling student behavior and knowledge significantly improves both student performance and next-question prediction and NANS is highly effective in selecting the informative negative samples.

Acknowledgements

This paper is based upon work supported by the National Science Foundation under Grant Number 2047500.

References

Chae, D.-K.; Kang, J.-S.; Kim, S.-W.; and Lee, J.-T. 2018. CFGAN: A generic collaborative filtering framework based on generative adversarial networks. In *international conference on information and knowledge management*, 137–146.

Chang, H.-S.; Hsu, H.-J.; and Chen, K.-T. 2015. Modeling Exercise Relationships in E-Learning: A Unified Approach. In *International Conference on Educational Data Mining*, 532–535.

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 *Learning at Scale*, 341–344.

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.

Chrysafiadi, K.; and Virvou, M. 2013. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11): 4715–4729.

CMU DataShop. 2015. Junyi dataset. https://pslcdatashop.web.cmu.edu/Project?id=244.

Dalipi, F.; Imran, A. S.; and Kastrati, Z. 2018. MOOC dropout prediction using machine learning techniques: Review and research challenges. In *EDUCON*, 1007–1014.

Désidéri, J.-A. 2012. Mutiple-gradient descent algorithm for multiobjective optimization. In *European Congress on Computational Methods in Applied Sciences and Engineering*.

Dwarampudi, M.; and Reddy, N. 2019. Effects of padding on LSTMs and CNNs. *arXiv preprint arXiv:1903.07288*.

Fan, L.; Pu, J.; Zhang, R.; and Wu, X.-M. 2023. Neighborhood-based Hard Negative Mining for Sequential Recommendation. In *SIGIR Conference on Research and Development in Information Retrieval*, 2042–2046.

Ghosh, A.; Heffernan, N.; and Lan, A. S. 2020. Context-Aware Attentive Knowledge Tracing. In *Conference on Knowledge Discovery & Data Mining*, 2330–2339.

Gillick, D.; Kulkarni, S.; Lansing, L.; Presta, A.; Baldridge, J.; Ie, E.; and Garcia-Olano, D. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*.

Goldberg, P.; Sümer, Ö.; Stürmer, K.; Wagner, W.; Göllner, R.; Gerjets, P.; Kasneci, E.; and Trautwein, U. 2021. Attentive or not? Toward a machine learning approach to assessing students' visible engagement in classroom instruction. *Educational Psychology Review*, 33: 27–49.

Guerra, J.; Sahebi, S.; Lin, Y.-R.; and Brusilovsky, P. 2014. The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises. In *International Conference on Educational Data Mining*.

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

Huang, T.; Dong, Y.; Ding, M.; Yang, Z.; Feng, W.; Wang, X.; and Tang, J. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Knowledge Discovery & Data Mining*, 665–674.

Kumar, M.; Singh, A.; and Handa, D. 2017. Literature survey on student's performance prediction in education using data mining techniques. *International Journal of Education and Management Engineering*, 7(6): 40–49.

Lin, X.; Zhen, H.-L.; Li, Z.; Zhang, Q.-F.; and Kwong, S. 2019. Pareto multi-task learning. *Advances in neural information processing systems*, 32.

Liu, Y.; Yang, X.; Zhou, S.; Liu, X.; Wang, Z.; Liang, K.; Tu, W.; Li, L.; Duan, J.; and Chen, C. 2023. Hard sample aware network for contrastive deep graph clustering. In *the AAAI conference on artificial intelligence*, volume 37, 8914–8922.

Lopez-del Rio, A.; Martin, M.; Perera-Lluna, A.; and Saidi, R. 2020. Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Scientific reports*, 10(1): 14634.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mirzaei, M. 2020. *Discriminative Factorization Models for Student Behavioral Pattern Detection and Classification*. State University of New York at Albany.

Pandey, S.; and Karypis, G. 2019. A Self-Attentive Model for Knowledge Tracing. In *International Conference on Educational Data Mining*, 384–389.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.; and Sohl-Dickstein, J. 2015. Deep Knowledge Tracing. In *Conference on Neural Information Processing Systems*, 505–513.

Pojen, C.; Mingen, H.; and Tzuyang, T. 2020. Junyi Academy Online Learning Activity Dataset: A large-scale public online learning activity dataset from elementary to senior high school students. *Dataset available from https://www.kaggle.com/junyiacademy/learning-activity-public-dataset-by-junyi-academy*.

Rendle, S.; and Freudenthaler, C. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *international conference on Web search and data mining*, 273–282.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Sahebi, S.; Lin, Y.-R.; and Brusilovsky, P. 2016. Tensor Factorization for Student Modeling and Performance Prediction in Unstructured Domain. *International Conference on Educational Data Mining*.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, 1842–1850.

Thai-Nghe, N.; Horváth, T.; and Schmidt-Thieme, L. 2010. Factorization models for forecasting student performance. In *International Conference on Educational Data Mining*.

Tornwall, J.; Lu, L.; and Xie, K. 2020. Frequency of participation in student response system activities as a predictor of final grade: An observational study. *Nurse Education Today*, 87: 104342.

VanLehn, K. 1988. Student modeling. Foundations of Intelligent Tutoring Systems. M. Polson and J. Richardson.

Wang, C.; Zhao, S.; and Sahebi, S. 2021. Learning from Non-Assessed Resources: Deep Multi-Type Knowledge Tracing. *International Conference on Educational Data Mining*.

Xia, J.; Wu, L.; Wang, G.; Chen, J.; and Li, S. Z. 2021. Progcl: Rethinking hard negative mining in graph contrastive learning. *arXiv preprint arXiv:2110.02027*.

Xiong, L.; Xiong, C.; Li, Y.; Tang, K.-F.; Liu, J.; Bennett, P.; Ahmed, J.; and Overwijk, A. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Yang, Z.; Ding, M.; Zhou, C.; Yang, H.; Zhou, J.; and Tang, J. 2020. Understanding negative sampling in graph representation learning. In *international conference on knowledge discovery & data mining*, 1666–1676.

Yang, Z.; Ding, M.; Zou, X.; Tang, J.; Xu, B.; Zhou, C.; and Yang, H. 2022. Region or global? a principle for negative sampling in graph-based recommendation. *Transactions on Knowledge and Data Engineering*, 35(6): 6264–6277.

Yeung, C. K. 2019. Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable using Item Response Theory. In *International Conference on Educational Data Mining*, 683–686.

Zahedi, L.; Lunn, S. J.; Pouyanfar, S.; Ross, M. S.; and Ohland, M. W. 2020. Leveraging machine learning techniques to analyze computing persistence in undergraduate programs. In *2020 ASEE Virtual Annual Conference Content Access*.

Zahedi, L. R. 2019. Implications of gamification in learning environments on computer sci-ence students: A comprehensive study. In *126th ICCE*.

Zhang, J.; Shi, X.; King, I.; and Yeung, D.-Y. 2017. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *International Conference on World Wide Web*, 765–774.

Zhang, W.; Chen, T.; Wang, J.; and Yu, Y. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR conference on Research and development in information retrieval*, 785–788.

Zhao, S.; and Sahebi, S. 2023. Graph-Enhanced Multi-Activity Knowledge Tracing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 529–546.

Zhao, S.; and Sahebi, S. 2024a. Exploring Simultaneous Knowledge and Behavior Tracing. In *International Conference on Educational Data Mining*, 927–932.

Zhao, S.; and Sahebi, S. 2024b. Multi-Task Modeling of Student Knowledge and Behavior. In *International Conference on Information and Knowledge Management*, 3363–3373.

Zhao, S.; and Sahebi, S. 2024c. Towards Multi-Objective Behavior and Knowledge Modeling in Students. In *User Modeling, Adaptation and Personalization*, 183–188.

Zhao, S.; Wang, C.; and Sahebi, S. 2020. Modeling Knowledge Acquisition from Multiple Learning Resource Types. In *International Conference on Educational Data Mining*, 313–324.

Zhao, S.; Wang, C.; and Sahebi, S. 2022. Transition-Aware Multi-Activity Knowledge Tracing. In *International Conference on Big Data*, 1760–1769.

Zhu, Y.; Xu, Y.; Cui, H.; Yang, C.; Liu, Q.; and Wu, S. 2021. Structure-aware hard negative mining for heterogeneous graph contrastive learning. *arXiv preprint arXiv:2108.13886*.

Zhu, Y.; Xu, Y.; Cui, H.; Yang, C.; Liu, Q.; and Wu, S. 2022. Structure-enhanced heterogeneous graph contrastive learning. In *SIAM International Conference on Data Mining*, 82–90.

Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Transactions on Evolutionary Computation*, 3(4): 257–271.