Exercise Recommendation Based on Feature-Aligned Knowledge Tracing

Wei Ji[0009−0003−4628−7270], Zhiyu Chen[0009−0004−5766−1691], and Jing Xiao(✉)[0000−0002−5242−7909]

School of Computer Science, South China Normal University, Guangzhou, China
xiaojing@scnu.edu.cn

Abstract. The rapid expansion of online education has provided students with a vast array of learning resources. However, this abundance has also led to knowledge overload and the challenge of selecting appropriate learning materials. To address this issue, exercise recommendation has become an essential strategy for supporting personalized learning. While traditional collaborative filtering and knowledge tracing (KT) methods offer viable solutions for personalized recommendations, they often fall short in fully leveraging the rich semantic information embedded within exercise texts. In this study, we introduce a feature-aligned knowledge tracing-based exercise recommendation method, called FAKT-ER, which integrates the semantic understanding power of large language models (LLMs) with deep knowledge tracing (DKT) techniques. By aligning semantic and collaborative data, FAKT-ER enables a more accurate representation of students' knowledge states, leading to improved exercise recommendations. Additionally, we incorporate feature dimensionality reduction and summation to enhance the model's computational efficiency and expressive capacity. Experimental results across three educational datasets demonstrate that FAKT-ER outperforms other methods in terms of recommendation diversity, novelty, and accuracy, significantly enhancing the students' online learning experience.

Keywords: Personalised exercise recommendation · Deep knowledge tracing · Feature alignment.

1 Introduction

The rise of online education has equipped students with a wealth of learning resources and flexible study schedules. However, it has also introduced challenges such as knowledge overload [17] and knowledge disorientation [32]. With an extensive array of educational materials available, students often struggle to quickly identify suitable content tailored to their needs. Exercises are a crucial component of the learning process, essential for reinforcing knowledge and evaluating learning outcomes. The rapid expansion of online exercises has further complicated students' ability to select appropriate ones. As a result, personalizing exercises to meet each student's needs within this vast pool of resources has become a significant focus in the realm of smart education.

To address this issue, various recommendation system technologies have been applied to exercise recommendations in recent years, making personalized exercise recommendation a significant research focus in intelligent education. Researchers treat students as users and exercises as products, considering students' responses to exercises as evaluations of these products. These methods enable the system to intelligently recommend the most suitable exercises based on students' learning progress and needs, thereby enhancing learning efficiency and optimizing the online education experience. Traditional exercise recommendation strategies, though structurally simple, often fail to account for students' knowledge mastery, resulting in less precise recommendations.

Knowledge Tracing (KT) is a technique used to dynamically tracks and assesses students' mastery of knowledge components (KCs). It offers personalized learning suggestions and resources by capturing explicit behavioral data such as students' accuracy in answering exercises, the time spent on each exercise, and the frequency of practice during the learning process. By continuously monitoring the state of students' implicit knowledge over time, KT infers their mastery of specific KCs and predicts their future performance on related topics. Figure. 1 presents a simplified schematic of the KT task. While KT-based methods can make recommendations by considering students' knowledge states, these methods are unable to utilize the rich semantic information contained in exercise texts and conceptual descriptions, resulting in lower prediction accuracy.

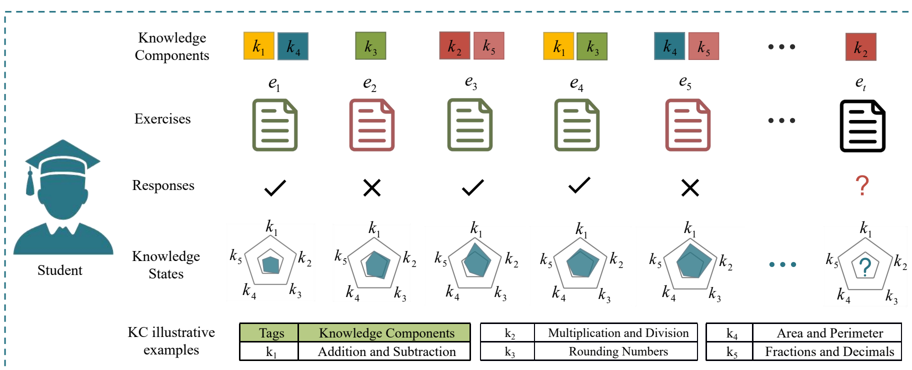

Fig. 1. A simple schematic of the KT task.

In this paper, we propose a feature-aligned knowledge-tracing based exercise recommendation method called FAKT-ER. This method aligns the semantic representations of students and exercises generated by a large language model (LLM) with collaborative data before inputting them into a model that performs knowledge tracing to dynamically model the students' knowledge states. Subsequently, it utilizes the collaborative filtering algorithm that combines the students' knowledge levels with the difficulty of the exercises to provide personalized exercise recommendations. To optimize model training, we design two loss functions to enhance the alignment of student and exercise features. Notably, we introduce a feature-enhanced deep knowledge tracing (FE-DKT) model, which applies feature dimensionality reduction and student's feature summation to the original deep knowledge tracing (DKT) task, thereby improving information sharing and interaction among features. Experimental results demonstrate that FAKT-ER significantly outperforms other baseline methods in recommendation diversity and novelty. Overall, our contributions can be summarized in the following three points:

(1) Introduction of the FE-DKT model: By applying feature reduction and student's feature summation to the traditional DKT model, we optimize feature utilization, integrate the student's abilities, and improve generalization, and enhance the model's effectiveness in capturing complex learning behaviors.

(2) Proposal of the FAKT-ER Method: This method aligns semantic representations generated by the LLM with collaborative data before feeding them into FE-DKT. The resulting knowledge level matrix is then combined with a collaborative filtering algorithm to deliver personalized exercise recommendations that match students' knowledge levels and exercise difficulty requirements.

(3) Design of the feature alignment loss function: To optimize feature alignment between students and exercises, we introduce a knowledge distillation loss and Bayesian personalized ranking loss, allowing the model to better capture the intrinsic relationship between students' knowledge mastery and exercises.

2 Related Work

In this section, we first review the research related to exercise recommendation. Then, we present a brief overview of the improvement of the recommendation model by the LLM.

2.1 Exercise Recommendation

With the rise of the internet and big data, personalized exercise recommendation has become a pivotal topic in educational data mining. This field has evolved from initial rule-based and statistical methods to incorporate collaborative filtering [6], content-based recommendations [3], and hybrid approaches [21], ultimately progressing to contemporary deep learning-driven recommendation systems [8, 22]. Existing researches have utilized a variety of methods to improve the accuracy and adaptability of personalized exercise recommendations.

Klašnja-Milićević et al.[10] and Wu et al.[26] analyzed students' historical behavior data to discern patterns in habits and interests, providing a basis for personalizing learning resource recommendations. Segal et al.[23] emphasized the design of teacher-assistive tools and individualized instruction, introducing the ranking algorithm EduRank. Saito et al.[20] proposed an exercise recommendation model that generated personalized suggestions by accounting for students' historical behavior, long-term learning objectives, and adaptive learning paths.

To address the challenges of collaborative filtering algorithms, such as cold-start and data sparsity, Li et al.[11] enhanced prediction accuracy and system generalization by combining improved spectral clustering with transfer learning. Similarly, Wang et al.[24] tackled the sparsity issue in scoring matrices by assessing student-exercise similarities, thereby delivering more accurate exercise recommendations.

However, relying solely on students' answer records often overlooks their actual knowledge levels, leading to recommendations for exercises that may be inappropriately difficult and potentially harming students' learning outcomes. While collaborative filtering algorithms are straightforward and easy to implement, they fail to adequately consider students' knowledge states, resulting in less rational recommendation outcomes. In contrast, knowledge modeling-based approaches can provide more accurate recommendations by dynamically assessing students' knowledge levels. For instance, Cheng et al.[2] introduced a simulated knowledge state update mechanism, while Wu et al.[28] employed recurrent neural network models to estimate students' mastery of knowledge components (KCs), both enabling a more precise assessment of students' knowledge levels and allowing for personalized exercise recommendations. Additionally, He et al. [9] enhanced recommendation accuracy by combining knowledge tracing with conceptual prerequisite relationships.

Although the knowledge modeling-based approach can dynamically assess knowledge levels, it does not fully account for the gradual and group nature of the learning process, leading to overly homogeneous recommendation results. Consequently, some studies combined DKT models with collaborative filtering algorithms to generate personalized exercise recommendations that aligned with a range of difficulty levels [13, 25]. Ma et al.[15] leveraged cognitive diagnostic techniques to evaluate learner states, applying the Neutrosophic set method for similarity quantification and integrating collaborative filtering to predict performance on new exercises, thus personalizing recommendations accordingly. Yan et al.[30] proposed a method that automatically generated a knowledge relation matrix using the Q-matrix, constructed a knowledge graph via DKT, and filtered exercises based on this graph, thereby improving the precision and impact of personalized recommendations.

2.2 LLM-Driven Recommendation Models

With significant breakthroughs in LLMs within the field of natural language processing (NLP), researchers began exploring the integration of LLMs with various deep learning tasks [18]. The extensive world knowledge, powerful semantic understanding, and contextual reasoning capabilities of LLMs offered new avenues for addressing complex reasoning tasks in recommendation systems [16, 27]. Xi et al.[29] explored the potential of LLMs to enhance user and item profile generation, resulting in more accurate recommendations. Li et al.[12] treated tabular data and transformed textual data as two distinct modalities, using recommendation models alongside pre-trained language models to achieve fine-grained alignment, integrating both collaborative and semantic insights. Zhu et al. [33] optimized a collaborative LLM specifically for recommendation tasks, applying a hybrid strategy that combined soft and hard prompting techniques. This approach facilitated the development of a generative recommender system that tightly integrated the LLM paradigm with the ID paradigm, allowing the model to effectively capture recommendation-related information from noisy user and item content.

3 Feature-Aligned Knowledge Tracing-Based Exercise Recommendation

This section proposes an innovative personalized exercise recommendation framework, the overall architecture of which is illustrated in Figure 2. The framework consists of three core modules: (1)the semantic representation module generates semantic vectors for students and exercises using the LLM, (2)the FE-DKT module takes these semantic representations, aligns them with collaborative data, and models students' knowledge states dynamically through the LSTM, and (3)the exercise recommendation module leverages collaborative filtering to suggest personalized exercises by integrating students' knowledge levels with problem difficulty.

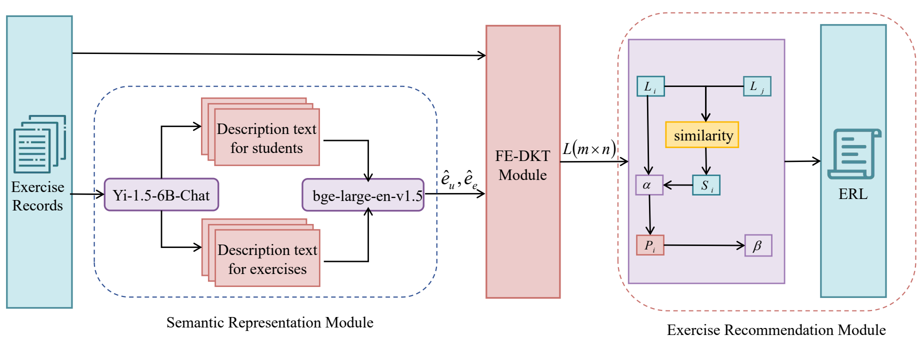

Fig. 2. Overall model architecture of the FAKT-ER. $\hat{e}_u$ and $\hat{e}_e$ represent the semantic embeddings of students and exercises, respectively. $L(m \times n)$ represents the knowledge level vector matrix, while L_i and L_j denote individual knowledge level vectors for student i and student j. S_i, P_i, α, and β represent similarity measure, exercise score vector, parameter, and exercise difficulty factor threshold, respectively.

3.1 Semantic Representation Module

This module produces descriptive text for both students and exercises by extracting the student's activity records and the occurrences of KCs in the exercises. The generated descriptions are then vectorized for final use in the FE-DKT model.

Text Description Generation Researchers [5] have emphasized that effective prompts should encompass sufficient semantic information regarding both students and exercises. For each student, we record the KCs associated with both correct and incorrect answers, integrating this information with the total number of exercises attempted, the number of correct responses, and the number of incorrect responses to create tailored prompts reflecting the student's behavior. For each exercise, we construct prompts based on the relevant KCs and the corresponding units. By inputting these prompts into Yi-1.5-6B-Chat [31], we generate descriptive information about both the students and the exercises.

Semantic Embedding We employ the bge-large-en-v1.5 [1] model as our primary embedding model, known for its strong capabilities in text embedding by transforming natural language into high-dimensional vectors for semantic similarity calculations. Utilizing this model, we encode the generated textual descriptions of students and exercises as fixed-length vectors, which facilitates feature alignment and supports subsequent recommendation tasks.

3.2 FE-DKT Module

This module predicts students' mastery of various KCs based on their historical answer records, utilizing an LSTM network to process the students' answer sequences.

Model Structure In the original DKT [19] model, the selection of feature dimensions can be redundant and underutilized. To optimize the model's performance, we reduce the original feature dimensions to half of their original size. Given a total of n exercises and m students, each with an embedding dimension d, the model structure of FE-DKT is shown in Figure 3.

The weight attribute of an Embedding layer is the layer's embedding matrix, which holds the vector representations corresponding to each ID. By default, these vectors are initialized randomly. However, in our model, instead of using randomly initialized embeddings, we use pre-generated semantic embeddings of students and exercises, created by the LLM. These semantic embeddings are used as weights during model training, allowing the model to leverage LLM-generated semantic information for learning and making recommendations.

In the input layer, the model utilizes the student sequence u, exercise sequence e and answer sequence a (where 0 denotes an incorrect response and 1 denotes a correct one). These sequences are converted into embedding vectors through the exercise embedding matrix $E \in \mathbf{R}^{n\times d}$, student embedding matrix $U \in \mathbf{R}^{m\times d}$, and answer embedding matrix $A \in \mathbf{R}^{2\times d}$, the student information is repeated at each time step. Rather than employing the simple concatenation of features as in the original DKT model, we introduce feature summation to create the feature vector x as input to the LSTM:

$$x = e_u + e_e + e_a \tag{1}$$

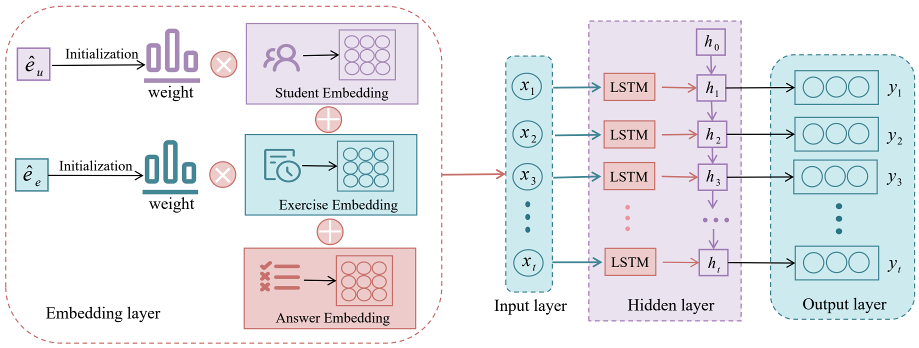

Fig. 3. Overall structure of the FE-DKT model. $\hat{e}_s$ and $\hat{e}_e$ denote the semantic embeddings of students and exercises, respectively. x_t denotes the input sequence of student behaviors at time t, and the vector y_t predicts the probability that the student will answer the practice question correctly by computing the hidden state h_t.

where e_u, e_e and e_a denote the student embedding, exercise embedding, and answer embedding, respectively.

The model generates a hidden state h_t representing the current knowledge state at each time step through an LSTM cell, which is subsequently used to compute the output vector y_t. The length of this vector corresponds to the number of exercises, with each element representing the probability of a student correctly answering a specific KC. The calculation of y_t is defined as:

$$y_t = \sigma(W_y h_t + b_y) \tag{2}$$

where W_y is weight matrice, b_y is the bias term, σ is the Sigmoid function. By training the DKT model, we compute the knowledge level vector for each student, resulting in a knowledge level vector matrix $L(m \times n)$ for all students.

Feature Alignment To optimize the model's ability to match personalized information, we introduce a Multi-Layer Perceptron (MLP) for further processing and transforming the semantic features of students and exercises. The MLP consists of two fully connected layers that map the embedded features into the appropriate representation spaces, allowing them to be merged with features from other components to generate a more refined feature representation. Let $\hat{e}_s$ and $\hat{e}_e$ denote the semantic embeddings of students and exercises, respectively. The mapping process can be expressed as:

$$f(\hat{e}) = \sigma\left(W_2 \cdot \sigma\left(W_1 \cdot \hat{e} + b_1\right) + b_2\right) \tag{3}$$

where W_1 and W_2 are weight matrices, b_1 and b_2 are bias terms, and σ is the Leaky ReLU activation function, $\hat{e}$ is the vector representation of the semantic embedding. The MLP-adjusted semantic embeddings are used in the computation of the comparative learning loss, assisting the model in aligning the semantic features of students and exercises during knowledge distillation. This alignment ensures that the model can better capture the intrinsic relationships between students' knowledge and the exercises, improving the overall recommendation accuracy.

3.3 Exercise Recommendation Module

The key step of this module is to calculate the cosine similarity of knowledge level vectors to identify students similar to the target students. For the knowledge level vector matrix L, we use the cosine similarity to compute the similarity matrix:

$$Sim(L_i, L_j) = \frac{L_i \cdot L_j}{\|L_i\|\|L_j\|} \tag{4}$$

where L_i and L_j denote the knowledge level vectors of the ith and jth students, respectively.

To avoid homogeneity of knowledge levels, the five users most similar to each student are selected in the similarity matrix and calibrated for mastery, and the final score vector for student i is updated as:

$$P_i = \alpha \times L_i + (1 - \alpha) \times S_i \tag{5}$$

where S_i is the mean vector of the knowledge levels of students similar to student i. The parameter α is used to adjust the influence of the individual student's knowledge level and the collective knowledge level of the student group on the recommendation results, with a value range of $[0, 1]$. The parameter α is set to 0.6 in this paper.

When recommending exercises, a predetermined exercise difficulty factor threshold β is employed to assess whether students have achieved a sufficient level of mastery for specific KCs. For instance, if the predicted value surpasses the threshold β, the corresponding exercises are deemed appropriate for recommendation for that KC. Ultimately, KCs that satisfy these criteria are included in the recommended list, enabling students to select the corresponding exercises for practice. In this paper, the threshold β is set to 0.7.

3.4 Design of Multitasking Loss Function

Binary Cross-Entropy Loss This loss function is used to optimize the prediction accuracy of students' mastery of KCs. Binary Cross-Entropy Loss calculates the difference between the probability of mastery predicted by the model and the actual outcome, defined as follows:

$$L_{BCE} = -\sum_{i=1}^{N} (t_i \log(y_i) + (1 - t_i) \log(1 - y_i)) \tag{6}$$

where t_i represents the true label (0 or 1) of the ith sample, indicating the user's response to the ith exercise. By minimizing this loss function, the model can gradually enhance the prediction accuracy of students' future performance.

Knowledge Distillation Loss we introduce a knowledge distillation loss based on contrastive learning. This approach utilizes information entropy to calculate the similarity difference between the semantic embeddings and the model's internal embeddings, ensuring the effectiveness of the alignment of the semantic features of both students and exercises:

$$L_{KD} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\frac{S(e_i,\hat{e}_i)}{\tau})}{\sum_{j=1}^{M}\exp(\frac{S(e_i,\hat{e}_j)}{\tau})} \tag{7}$$

where N denotes the batch size, M represents the number of comparison samples, e_i, $\hat{e}_i$, and $\hat{e}_j$ represent the original embedding vectors of the model, the embedding vectors obtained through MLP mapping, and the embedding vectors in the entire set, respectively. The function $S(\cdot,\cdot)$ measures the similarity between two vectors, and τ is the temperature parameter, which regulates the differences between the embeddings.

Bayesian Personalized Ranking Loss This loss function emphasizes the relative preferences of students for positive samples (exercises answered correctly) and negative samples (exercises answered incorrectly). The core idea is to maximize the difference between the students' ratings of positive samples and their ratings of negative samples:

$$L_{BPR} = -\sum_{i=1}^{N}\log\sigma(y_i^+ - y_i^-) \tag{8}$$

where y_i^+ and y_i^- represent the prediction scores of student i for positive and negative samples, respectively. The BPR loss encourages the model to learn meaningful user preferences, thereby improving the quality of recommendations.

Finally, a composite loss function can be constructed:

$$L = L_{BCE} + \lambda_1 L_{BPR} + \lambda_2 L_{KD} \tag{9}$$

where λ_1 and λ_2 are hyperparameters used to balance the contributions of each loss function to the overall loss.

4 EXPERIMENT

In this section, we conduct extensive experiments to evaluate the recommendation performance of our approach across different datasets.

4.1 Datasets

We evaluate the effectiveness of our method on three datasets: Algebra2005, ASSIST2009, and ASSIST2015. Inspired by pyKT [14], we perform the following preprocessing steps on the data: (1) records with missing KC names are removed;

(2) duplicates are removed from both the order of responses column and the KC name column to retain unique response records. To maintain the chronological order of the data, all records are sorted by the order of responses; (3) a length matching operation is conducted on exercise and response sequences in accordance with model requirements. Specifically, sequences exceeding the specified length are truncated to a fixed length, while shorter sequences are padded with -1 to ensure uniform sequence lengths. A detailed overview of these datasets is presented in Table 1.

Table 1. Detailed overview of the preprocessed datasets.

Datasets	Students	Exercises	KCs	Records	R per S
Algebra2005	574	1,084	436	607,025	1057.53
ASSIST2009	4,151	16,891	110	325,637	78.44
ASSIST2015	19,840	562	421	623,801	31.44

Algebra2005: This dataset is presented at the KDDCup2010 Educational Data Mining Challenge and contains data on learners' responses to algebra exercises collected during the 2005-2006 academic year, extracted from the intelligent tutoring system Cognitive Tutors.

ASSISTment: The dataset is a widely used educational dataset in the field of educational data mining. It includes over 100,000 pieces of student answer data collected from online tutoring systems for elementary school math courses, encompassing question sequences, student information, and additional data. The datasets used in this experiment are ASSIST2009 and ASSIST2015. ASSIST2009 contains a wealth of valuable auxiliary information, where as ASSIST2015 lacks information about the exercises and other supplementary data. Although ASSIST2015 has more records overall compared to ASSIST2009, the average number of records per student is smaller due to the larger number of students.

4.2 Baselines

To evaluate the effectiveness of the recommendation strategies proposed in this paper, we conduct comparison experiments with multiple exercise recommendation strategies across each of the three datasets. The following base models are selected for these comparison experiments:

User-CF [4]: It identifies similar student groups using cosine similarity and predicts target students' performance on unanswered exercises based on these similarities, recommending the most likely correct answers.

DKT [19]: By tracing students' answer sequences, RNNs are used to dynamically update the students' knowledge states. Based on the student's current knowledge state, the model predicts the likelihood of correct answers on unanswered exercises and recommends exercises where the student is expected to have a higher success rate.

DKT-CF [25]: The DKT model generates a knowledge state vector for each student and uses collaborative filtering to find similar users. By averaging their scores, the model predicts the target student's performance and recommends exercises of suitable difficulty.

DKT-SVD++ [7]: This method uses the DKT model to generate each student's knowledge state matrix, which is then enhanced through matrix decomposition to predict scores for unanswered exercises, ultimately recommending exercises that align with the student's knowledge level.

4.3 Experimental Setup

We train the FE-DKT model using the AdamW optimizer, setting the learning rate to 0.001. The model employs 100 LSTM nodes in the hidden layer, with the training process configured for 20 epochs, a batch size of 256, and a sequence length of 100. To mitigate overfitting, dropout is applied. For all three datasets, 80% of the students are randomly divided into training and validation sets, while the remaining 20% are reserved for testing the recommendation methods.

4.4 Evaluation metrics

In this experiment, FAKT-ER is compared with established classical KT models. The performance of the models is evaluated using three metrics: accuracy (ACC), area under the ROC curve (AUC), and root mean square error (RMSE).

In order to measure the effectiveness of the method proposed in this paper to generate personalised exercise recommendations to students, we adopts six indicators, namely Precision, Recall, F1 value, Accuracy, Novelty and Diversity, to evaluate and compare the effectiveness of different exercise recommendation strategies.

In recommender systems, novelty and diversity are two crucial attributes for assessing the quality of recommendations [7]. Novelty typically refers to the innovation and rarity of the recommendations, while diversity pertains to the breadth and variability of the recommended content. These concepts have distinct meanings and applications depending on the domain of the recommender system. For the purposes of this experiment, we define novelty and diversity as follows:

Novelty: The novelty of the exercise recommendations is assessed by calculating the cosine similarity between the recommended exercises and the exercises previously done by the students. The novelty of the list of exercise recommendations $R(u)$ for student u is defined as follows:

$$Novelty = 1 - \frac{1}{|M|} \sum_{u \in M} \frac{|Q(u) \cap R(u)|}{|Q(u)|} \tag{10}$$

where M represents the set of all recommended students, and $Q(u)$ denotes the set of unfinished exercises for student u.

Diversity: In exercise recommendation, diversity measures the differences among the recommended exercises. This paper uses Euclidean distance to quantify the similarity between exercises. The diversity of the list of exercise recommendations $R(u)$ for student u is defined as follows:

$$Diversity(R(u)) = \frac{\sum_{i,j \in R(u),\, i \neq j} dist(e_i, e_j)}{|R(u)|(|R(u)| - 1)} \tag{11}$$

where $dist(e_i, e_j)$ represents the euclidean distance between the embedding vectors of exercises e_i and e_j.

5 Results

5.1 Model Predictive Performance

To evaluate the predictive performance of our proposed model, FAKT-ER, we compare it with baseline methods, including the DKT and FE-DKT models. As shown in Table 2, FAKT-ER consistently surpasses these baseline models across all three datasets in terms of AUC, ACC, and RMSE, highlighting its effectiveness in accurately predicting student performance.

In the comparison of models, the FE-DKT model performed better than the DKT model, which suggests that capturing the intrinsic relationships between features through dimensionality reduction and weighting can enhance the performance of the KT model. FAKT-ER outperforms the FE-DKT model, demonstrating that aligning synergistic relational representations with semantic representations enhances the performance of KT models.

Table 2. Comparison of model prediction performance.

Models	Algebra2005			ASSIST2009			ASSIST2015		
	AUC	ACC	RMSE	AUC	ACC	RMSE	AUC	ACC	RMSE
DKT	0.7378	0.7833	0.4654	0.7250	0.7269	0.5225	0.7119	0.7591	0.4907
FE-DKT	0.8236	0.8183	0.4261	0.8145	0.7778	0.4713	0.7387	0.7696	0.4799
FAKT-ER	**0.8917**	**0.8264**	**0.4165**	**0.8706**	**0.7999**	**0.4473**	**0.8693**	**0.8069**	**0.4394**

5.2 Exercise Recommendation Performance

Table 3 displays the results of evaluating the six methods based on the metrics of precision, recall, and F1 score. In experiments conducted on three datasets, our proposed exercise recommendation method outperforms other algorithms, achieving the highest precision, recall, and F1 scores. This demonstrates that FAKT-ER yields more accurate recommendation results and is more effective than the alternatives.

Table 3. Comparison of exercise recommendation performance, where P denotes precision and R denotes recall.

Models	Algebra2005			ASSIST2009			ASSIST2015		
	P	R	F1	P	R	F1	P	R	F1
User-CF	0.5151	0.2480	0.2986	0.2909	0.2718	0.2744	0.3939	0.5187	0.4457
DKT	0.82	0.7387	0.7772	0.7049	0.6174	0.6583	0.6818	0.7894	0.7317
FE-DKT	0.8447	0.8333	0.8390	0.7286	0.8020	0.7635	0.7127	0.6789	0.6954
DKT-CF	0.8371	0.8416	0.8393	0.7305	0.8187	0.7721	0.7194	0.6815	0.7
DKT-SVD++	0.8247	0.8693	0.8464	0.7286	0.8288	0.7755	0.7002	0.7315	0.7155
FAKT-ER	**0.8536**	**0.8750**	**0.8641**	**0.7752**	**0.8414**	**0.8070**	**0.7818**	**0.8113**	**0.7962**

The comparison results of accuracy, novelty, and diversity in the exercise recommendation lists are shown in Tables 4–6. The Mean reflects the average performance of each indicator in the recommendation list, with values closer to 1 indicating better performance. The standard deviation (Std.D) represents the variability of each indicator, where values closer to 0 suggest more stable recommendation outcomes and improved overall performance. Observe the data presented in Tables 4–6:

(1) FAKT-ER outperforms the baseline model in accuracy, novelty, and diversity across all three datasets, and its validity is even more stable than that of the baseline model.

(2) User-CF lacks the ability to dynamically update because it only considers static feature representations, ignoring the specificity of the education domain. As a result, User-CF shows a significant performance gap across all three metrics compared to other baseline models.

(3) Novelty is related to the average number of records per student. In the ASSIST2015 dataset, the recommendation lists show higher novelty due to a larger number of students and a smaller average number of records per student.

(4) The diversity is related to the size of the dataset. For example, in the ASSIST2009 dataset, due to the large number of exercises and fewer types of topics, the diversity of the generated recommendation lists is lower. In the ASSIST2015 dataset, although the number of exercises is smaller, the diversity of the recommendation lists is higher because of the greater variety of topics.

Table 4. Comparison of accuracy indicators.

	Algebra2005		ASSIST2009		ASSIST2015	
	Mean	Std.D	Mean	Std.D	Mean	Std.D
User-CF	0.5080	0.0371	0.3337	0.0533	0.3626	0.0341
DKT	0.7465	0.0264	0.5782	0.0321	0.5809	0.0335
FE-DKT	0.7601	0.0164	0.6403	0.0142	0.5821	0.0448
DKT-CF	0.7605	0.0132	0.7002	0.0274	0.6579	0.0453
DKT-SVD++	0.7722	0.0079	0.7027	0.0354	0.6894	0.0489
FAKT-ER	**0.7845**	**0.0069**	**0.7642**	**0.0051**	**0.7007**	**0.0226**

Table 5. Comparison of novelty indicators.

	Algebra2005		ASSIST2009		ASSIST2015	
	Mean	Std.D	Mean	Std.D	Mean	Std.D
User-CF	0.6067	0.0996	0.2867	0.1439	0.2029	0.1921
DKT	0.7354	0.2072	0.5580	0.1467	0.7372	0.3103
FE-DKT	0.7650	0.0362	0.6639	0.0935	0.6536	0.2090
DKT-CF	0.8887	0.0544	0.8695	0.0943	0.8826	0.2183
DKT-SVD++	0.8931	0.0508	0.9009	0.0812	0.9170	0.1472
FAKT-ER	**0.9076**	**0.0244**	**0.9367**	**0.0388**	**0.9403**	**0.0330**

Table 6. Comparison of diversity indicators.

	Algebra2005		ASSIST2009		ASSIST2015	
	Mean	Std.D	Mean	Std.D	Mean	Std.D
User-CF	0.1816	0.0212	0.2230	0.0154	0.1113	0.0224
DKT	0.5877	0.0451	0.6930	0.0510	0.3496	0.0212
FE-DKT	0.5516	0.0266	0.6783	0.0464	0.4714	0.1066
DKT-CF	0.6231	0.0433	0.6951	0.0643	0.6333	0.1651
DKT-SVD++	0.7174	0.0732	0.7121	0.0637	0.7730	0.0987
FAKT-ER	**0.8125**	**0.0102**	**0.7686**	**0.0111**	**0.8529**	**0.0186**

6 Conclusion

In this paper, we propose an exercise recommendation method based on feature-aligned knowledge tracing, called FAKT-ER. By employing feature reduction and student's feature summation techniques, we effectively enhance the model's generalization ability and improve the dynamic understanding of student knowledge acquisition. This is achieved by aligning the semantic information generated by the LLM with the collaborative information from the FE-DKT model. Our experiments demonstrate that the recommended exercise lists generated by FAKT-ER across three real-world datasets exhibit greater diversity and novelty, indicating that our method can effectively address the learning needs of various students.

In future work, we aim to extend FAKT-ER by incorporating multimodal data, such as visual and auditory inputs, to further enrich the understanding of students' learning behaviors and preferences. Additionally, exploring adaptive feature alignment techniques to better capture student behavior patterns and improve the model's adaptability to diverse educational contexts.

Acknowledgments. This paper is supported by the National Natural Science Foundation of China No.62177015.

References

1. Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216 (2024)
2. Cheng, B., Zhang, Y., Shi, D.: Ontology-based personalized learning path recommendation for course learning. In: 2018 9th International Conference on Information Technology in Medicine and Education (ITME). pp. 531–535. IEEE (2018)
3. De Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. Recommender systems handbook pp. 119–159 (2015)
4. Feng, J.: Design and implementation of exercise system based on collaborative filtering algorithm(in chinese). Electronic Technology and Software Engineering pp. 67–69 (2020)
5. Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In: Proceedings of the 16th ACM Conference on Recommender Systems. pp. 299–315 (2022)
6. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM **35**(12), 61–70 (1992)
7. Guo, Y., Wang, M., Xiao, M.: Exercise recommendation method combining deep knowledge tracing and matrix completion(in chinese). Computer Technology and Development **33**(07), 188–195 (2023)
8. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. pp. 173–182 (2017)
9. He, Y., Wang, H., Pan, Y., Zhou, Y., Sun, G.: Exercise recommendation method based on knowledge tracing and concept prerequisite relations. CCF Transactions on Pervasive Computing and Interaction **4**(4), 452–464 (2022)
10. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: E-learning personalization based on hybrid recommendation strategy and learning style identification. Computers & education **56**(3), 885–899 (2011)
11. Li, X., Wang, Z., Hu, R., Zhu, Q., Wang, L.: Recommendation algorithm based on improved spectral clustering and transfer learning. Pattern Analysis and Applications **22**, 633–647 (2019)
12. Li, X., Chen, B., Hou, L., Tang, R.: Ctrl: Connect collaborative and language model for ctr prediction. arXiv preprint arXiv:2306.02841 (2023)
13. Liu, Z., Li, Y., Wei, L., Wang, W.: Adaptive exercise recommendation based on cognitive level and collaborative filtering. In: International Conference on Computer Science and Education. pp. 503–518. Springer (2022)
14. Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., Luo, W.: pykt: a python library to benchmark deep learning based knowledge tracing models. Advances in Neural Information Processing Systems **35**, 18542–18555 (2022)
15. Ma, H., Huang, Z., Tang, W., Zhang, X.: Exercise recommendation based on cognitive diagnosis and neutrosophic set. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 1467–1472. IEEE (2022)
16. Mysore, S., McCallum, A., Zamani, H.: Large language model augmented narrative driven recommendations. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 777–783 (2023)

17. Nabizadeh, A.H., Leal, J.P., Rafsanjani, H.N., Shah, R.R.: Learning path personalization and recommendation methods: A survey of the state-of-the-art. Expert Systems with Applications **159**, 113596 (2020)
18. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
19. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28** (2015)
20. Saito, T., Watanobe, Y.: Learning path recommender system based on recurrent neural network. In: 2018 9th international conference on awareness science and technology (iCAST). pp. 324–329. IEEE (2018)
21. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 253–260 (2002)
22. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web. pp. 111–112 (2015)
23. Segal, A., Katzir, Z., Gal, K., Shani, G., Shapira, B.: Edurank: A collaborative filtering approach to personalization in e-learning. In: Educational Data Mining 2014. Citeseer (2014)
24. Wang, H., Fu, W.: Personalized learning resource recommendation method based on dynamic collaborative filtering. Mobile Networks and Applications **26**, 473–487 (2021)
25. Wang, Z., Yu, N.: Education data-driven online course optimization mechanism for college student. Mobile Information Systems **2021**(1), 5545621 (2021)
26. Wu, D., Lu, J., Zhang, G.: A fuzzy tree matching-based personalized e-learning recommender system. IEEE transactions on fuzzy systems **23**(6), 2412–2426 (2015)
27. Wu, L., Qiu, Z., Zheng, Z., Zhu, H., Chen, E.: Exploring large language model for graph data understanding in online job recommendations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 9178–9186 (2024)
28. Wu, Z., Li, M., Tang, Y., Liang, Q.: Exercise recommendation based on knowledge concept prediction. Knowledge-Based Systems **210**, 106481 (2020)
29. Xi, Y., Liu, W., Lin, J., Cai, X., Zhu, H., Zhu, J., Chen, B., Tang, R., Zhang, W., Zhang, R., et al.: Towards open-world recommendation with knowledge augmentation from large language models. arXiv preprint arXiv:2306.10933 (2023)
30. Yan, Z., Du, H., Lin, Z., Jianhua, Z.: Personalization exercise recommendation framework based on knowledge concept graph. Computer Science and Information Systems **20**(2), 857–878 (2023)
31. Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., Li, H., Zhu, J., Chen, J., Chang, J., et al.: Yi: Open foundation models by 01. ai. arXiv preprint arXiv:2403.04652 (2024)
32. Zhong, L., Wei, Y., Yao, H., Deng, W., Wang, Z., Tong, M.: Review of deep learning-based personalized learning recommendation. In: Proceedings of the 2020 11th International conference on E-education, E-business, E-management, and E-learning. pp. 145–149 (2020)
33. Zhu, Y., Wu, L., Guo, Q., Hong, L., Li, J.: Collaborative large language model for recommender systems. In: Proceedings of the ACM on Web Conference 2024. pp. 3162–3172 (2024)