GEPR: Group-knowledge Enhanced Personalized Educational Resource Recommendation

LIQIN WANG, HANSHUO LIU, RUI ZHANG, XU WANG, and YONGFENG DONG*, Hebei University of Technology, China

To address the issues of sparse user interaction data and insufficient mining of knowledge features from the perspective of single users in online education platforms, a group-knowledge enhanced personalized educational resource recommendation model (GEPR) is proposed.First, a user-resource interaction graph is constructed based on multi-user interaction sequences, and user groups are iteratively partitioned through behavioral features.Next, the complex associations between resources and knowledge points are mined from text information, and a semantic information constraint term is designed to optimize the knowledge representation learning of resource entities, enhancing resource representation in both structural and textual dimensions.Then, a cross-attention mechanism with group knowledge constraints is designed to establish group-individual knowledge propagation channels, achieving the fusion of group and individual knowledge features through adaptive adjustment of attention weights.Finally, group knowledge information is integrated into the state representation and reward function of reinforcement learning (RL)to optimize the recommendation strategy. Comparative experiments with nine recommendation methods on two online education datasets show that GEPR can fully combine the semantic information of resources and group knowledge information, capture users' personalized needs, and improve recommendation accuracy.

JAIR Associate Editor: Viviana Mascardi

JAIR Reference Format:
Liqin Wang, Hanshuo Liu, Rui Zhang, Xu Wang, and Yongfeng Dong. 2026. GEPR: Group-knowledge Enhanced Personalized Educational Resource Recommendation. *Journal of Artificial Intelligence Research* 86, Article 34 (July 2026), 24 pages. DOI: 10.1613/jair.1.21303

1 Introduction

In the digital education era, online education platforms are vital for learners. Nevertheless, they often encounter the challenge of sparse user interaction data, which hinders the performance of educational resource recommendation systems. Existing recommendation methods have certain limitations. For instance, GRU4Rec's sequence modeling framework relies solely on item atomic ID embeddings, neglecting the structured semantic associations of knowledge graphs and multimodal knowledge. SASRec, a self-attention model, struggles to capture high-frequency user behavior information, affecting the precise grasp of short-term user interests. LightGCN, utilizing traditional natural language processing techniques, directly obtains embedding vectors from text to aid recommendations but fails to thoroughly explore complex correlations between user and knowledge state multi-segment related texts, resulting in shallow representations and an inadequate understanding of user knowledge preferences. Moreover, in data-sparse scenarios, limited user behavior data makes it difficult to comprehensively and accurately depict user knowledge features from a single user perspective, impacting personalized educational resource

*Corresponding Author.

Authors' Contact Information: Liqin Wang, ORCID: 0000-0002-9692-7833, wangliqin@hebut.edu.cn; Hanshuo Liu, ORCID: 0009-0008-9617-0368, liuhanshuo1998@126.com; Rui Zhang, 924564281@qq.com; Xu Wang, wxu0730@126.com; Yongfeng Dong, dongyf@hebut.edu.cn, Hebei University of Technology, Tianjin, China.

DOI: 10.1613/jair.1.21303

recommendation. However, implicit knowledge within similar user groups can supplement individual user knowledge features. To tackle these issues, this paper proposes GEPR. It deeply mines resource structure and semantics, introduces user group knowledge features, and optimizes the state representation and reward design of RL agents. This enhances the learning of user personalized knowledge features, fully explores the deep knowledge features of educational resources, and leverages group knowledge to compensate for the insufficiency of single-user knowledge features, thereby improving the accuracy and personalization of educational resource recommendations. The main contributions of this paper are as follows:

- A text semantic association constraint item is designed to optimize the knowledge graph representation learning process, effectively integrating the graph structure and text semantic information into the enhancement of resource representation.
- A cross-attention mechanism limited to group knowledge is constructed to dynamically fuse group and user knowledge, and group knowledge is incorporated into the representation of the RL state and the joint reward function. This uses group knowledge to supplement the knowledge of behavior-sparse users, optimizes recommendation strategies, and increases the performance of personalized educational resource recommendations. for posting by the author.
- Experimental results on the MOOCCube and MOOPer datasets show that GEPR outperforms existing recommendation methods in multiple evaluation metrics, particularly excelling in data-sparse scenarios.

2 Related Work

To enhance learning experiences in the booming online education sector, educational resource recommendations play a crucial role. Current recommendation methods mainly focus on mining user behavior data and resource content information to achieve precise educational resource recommendations. Recommendation algorithms can be categorized into sequence-based, graph-based, and hybrid information-based approaches.

2.1 Sequential Recommendation Methods

In the early research of recommendation technologies, Collaborative Filtering (CF) (Goldberg et al. 1992) served as one of the core methods to identify users' interested information . Its basic idea is to predict users' interest in unknown items by leveraging their historical behaviors or the behaviors of similar users. With the increase in modern data scale and complexity, the rich features in datasets have led to a significant expansion of data dimensions. In high-dimensional data scenarios, traditional CF methods suffer from exponential growth in computational complexity and limitations in handling complex user behaviors. As recommendation systems evolved, researchers realized that user behaviors are not isolated but exhibit sequential dependency. To model such dependencies, more complex and dynamic recommendation methods were explored. The Markov Chain (MC), as a method describing state transitions, has inherent advantages in predicting users' future behaviors. From CF to MC applications, recommendation systems have evolved from static to dynamic and from single to diverse. With the deepening of deep learning technologies, neural networks have significantly promoted performance improvement in various fields. Compared with traditional machine learning methods, neural networks possess more powerful feature extraction and representation learning capabilities, demonstrating unique advantages in processing large-scale, high-complexity long-sequence data. The Recurrent Neural Network (RNN), a neural network model suitable for sequential data, introduces cyclic connections to enable the network to memorize and utilize historical state information, thereby better understanding complex patterns in sequences.

Although RNN performs well in processing sequential data, the problems of gradient vanishing and explosion become prominent when sequences are long. This has led to improvements such as the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Cho et al. 2014). The introduction of gating mechanisms addresses the defects of traditional RNN, enabling models to handle long sequences and

capture long-term dependencies. For example, the GRU for Recommendation (GRU4Rec) (Hidasi et al. 2016) uses GRU to process sequential data and introduces a ranking loss function suitable for recommendation tasks to optimize the ranking of recommendation targets in the list. To break through the fixed window size limitation of traditional RNN and handle longer sequences, Transformer (Vaswani et al. 2017) and BERT (Devlin et al. 2019) (Bidirectional Encoder Representations from Transformers) introduce self-attention mechanisms, allowing models to acquire global information in sequences and significantly enhancing long-sequence processing capabilities. The Self-attention based Sequential Recommendation Model (SASRec) (Kang and McAuley 2018) emerges to capture long-term semantics and predict based on relatively few actions, introducing self-attention to the sequential recommendation field for the first time and using a double-layer Transformer decoder to capture users' sequential behaviors. The Time Interval aware Self-Attention based sequential recommendation (TiSASRec) (Li et al. 2020) considers the impact of interaction time, introduces relative time interval encoding to explicitly model resource interaction time features , and jointly captures absolute position information and relative time intervals in attention mechanisms to comprehensively capture dependencies in sequences, enhancing the model's ability to capture dynamic user behavior features. The Frequency Enhanced Hybrid Attention Network (FEARec) (Du et al. 2023) points out that self-attention mechanisms typically act as low-pass filters , making it difficult to capture high-frequency information, and previous methods struggle to distinguish periodic user behaviors in the time domain. The model uses Fourier transform to convert periodic trends of user behaviors from time-domain features to frequency-domain, designs a frequency-domain self-attention mechanism to capture periodic behavior features, and optimizes recommendation performance. However, sequential feature-based recommendation algorithms often face the problem of sparse interaction data, especially in recommendation tasks for long-tail items or new users, where data sparsity is more severe, making it difficult for models to construct accurate user behavior models. Additionally, sequential information typically focuses on individual user behavior patterns while ignoring macro-global information, such as group behavior trends or global correlations between resources. To address these issues, the idea of using graph structures to build complex connections has been introduced into recommendation algorithms.

2.2 Graph-based Recommendation Methods

In the digital era, user interaction data from social platforms, e-commerce, and other network applications exhibits high complexity, revealing not only interaction behavior features between users and resources but also deep information about social connections between users and associations between resources. As an efficient data representation tool, graph structures can intuitively represent users, resources, and their interaction relationships. The core of graph structure-based recommendation algorithms is to represent users and resources as nodes in a graph, model user-resource interactions as edges, and generate recommendations by mining graph structure features. Graph structures integrate fragmented data, comprehensively reveal internal laws and relationship patterns, and effectively alleviate data sparsity.

Random Walk (Paudel and Bernstein 2021), a graph-based traversal method describing the process of randomly selecting neighbor nodes in a graph or network, can simulate users' behaviors in resource networks for recommendation systems and generate results based on behavior probabilities. However, Random Walk has limited global information capture capabilities and weak scalability and generalization. In recent years, with the development of Graph Neural Networks (GNN), their powerful graph structure modeling capabilities have been widely applied to recommendation systems. GNN propagates information between nodes to comprehensively capture implicit connections in user behaviors, enabling precise personalized recommendations. He et al. (He et al. 2020) designed a simplified GCN model, LightGCN, which retains key components such as neighborhood aggregation and multi-layer propagation in GCN. Through multi-layer graph convolution operations, LightGCN propagates and

aggregates information on user-item interaction graphs, and the weighted summation of multi-layer user and item embeddings allows it to capture higher-order information, thereby improving recommendation performance.

Knowledge Graph (KG), as a structured knowledge representation, injects rich semantic information into recommendation systems. Typically, KG-integrated recommendation algorithms link resource nodes in interaction graphs with knowledge nodes in KGs to construct more complex graph structures. The Knowledge Graph Convolution Network (KGCN) (H. Wang et al. 2019), based on graph convolution, mines deep associations between users and resources through multi-layer neighborhood sampling and aggregation. Wang et al. proposed the Knowledge Graph Attention Network (KGAT) (X. Wang et al. 2019), which uses Graph Attention Network (GAT) to propagate relevant information in KGs.

Graph structure-based recommendation methods effectively enhance the ability to model users' personalized features by modeling global associations between users and resources. Meanwhile, the introduction of KGs enriches the information sources of recommendation systems, significantly improving cold-start issues caused by data sparsity and enhancing model robustness and generalization.

2.3 Hybrid Information-based Recommendation Methods

With the rapid development of Internet technologies, the data types and information dimensions involved in recommendation tasks have become increasingly complex, covering multiple forms such as text, images, and videos. These data contain diverse knowledge features, providing sufficient information support for recommendation system design. To deepen the understanding and prediction of user preferences, relying on single-type data is insufficient. Therefore, integrating multiple data types such as graph structures and sequential patterns has become a crucial strategy for recommendation algorithm design and optimization.

Graph structures play a key role in revealing complex relationships among users-resources, resources-resources, and users-users, while user behavior sequences contain abundant personalized behavior patterns and preference change trends. Hybrid information-based recommendation methods can comprehensively utilize multiple data structures and sources, not only alleviating data sparsity issues in single data sources but also capturing interactions and correlations among various information, thereby deeply mining user preferences and resource features to improve recommendation coverage and accuracy. To learn high-order dependencies between entities and excavate hidden association patterns in data, the Knowledge Graph Attention Network Enhanced Sequential Recommendation (KGSR) (X. Zhu et al. 2020) introduces high-order relationships between KG entities into sequential recommendation, uses graph attention networks to update node embeddings, obtains multi-layer connection information from higher-order neighbor nodes through stacked propagation layers, and uses GRU to model user interaction sequences, revealing potential patterns hidden in data and enhancing the model's understanding of users' dynamic preferences. Zhu et al. (T. Zhu et al. 2023) combine sequential relationships in user behavior sequences with semantic relationships in resource attributes to construct hybrid resource graphs, generate smooth resource embeddings through graph convolution operations, and use them as inputs for sequential recommendation models, thereby enhancing the model's ability to capture global resource features . Considering the advantages of RL in adapting to dynamic scenarios, Wang et al. (P. Wang et al. 2020) enhance information input in the decision-making process of intelligent agents based on KGs and sequential features, predict users' future knowledge preferences using graph structure information, and adjust recommendation strategies by constructing joint rewards at the sequence and knowledge levels . The Multi-knowledge Enhanced Graph Convolution (MkEGC) (Dong et al. 2024) further explores the joint representation of graph and sequential information, establishes dual graph convolution neural networks enhanced by resource-knowledge and user-knowledge, enhances user representation from multi-angle interaction perspectives, and optimizes educational resource recommendation performance.

In summary, hybrid information-based recommendation methods integrate multiple data types and sources, analyze user needs from multi-dimensional and multi-level perspectives, and significantly improve recommendation coverage, accuracy, and adaptability. This combination of multi-source data fusion and deep modeling provides strong support for diverse recommendation scenarios such as education, e-commerce, and social media. However, existing methods still have limitations in addressing data-sparse scenarios, particularly in mining deep knowledge features of educational resources and fusing group knowledge. In response to these challenges, this paper proposes a Group Knowledge-Enhanced Personalized Educational Resource Recommendation model (GEPR). GEPR constructs user-resource interaction graphs, divides user groups, and uses group knowledge to compensate for the insufficiency of individual user knowledge features. Meanwhile, it designs a knowledge association semantic representation module to deeply mine semantic correlations between resources and knowledge points, introduces a cross-attention mechanism to dynamically fuse group knowledge and user knowledge, and optimizes recommendation strategies.

3 Methodology

GEPR fully mines the dual features of resource text and graph structure and uses group knowledge to dynamically supplement user personalized knowledge modeling. The overall framework of GEPR is shown in Figure 1 and includes three modules: a user group identification module based on behavioral features, a knowledge association semantic representation module, and a group knowledge constrained recommendation module.

3.1 User Group Identification Module Based on Behavioral Features

In recommendation systems, users who interact with the same resources can be closely linked. For example, users A and B who have both watched the teaching video "Computer Organization Principles" may have similar learning needs or goals. This study draws on the classic Leiden algorithm (Traag et al. 2019) to divide groups based on the connection patterns and density between user and resource nodes. First, all user-resource interactions are constructed into an interaction behavior graph, where nodes represent all users U and resources I, and edges represent all user-resource interactions. Then, each node is initially treated as a separate group. The algorithm randomly selects a node and transfer it to another group, updating the quality function as shown in formula (1).

$$Q = \frac{1}{2m} \sum_{C} \left[e_c - \mu \frac{k_i k_j}{2m} \right] \tag{1}$$

where m is the total number of edges in the interaction graph, e_c is the number of edges within group C, k_i and k_j are the degrees of nodes i and j, and $\frac{k_i k_j}{2m}$ is the expected number of edges within the group. μ is the resolution parameter that determines the granularity of group division. When node movement increases Q, the node is allocated to group g. Next, for each initially divided group, internal nodes are refined to separate low-connection rate nodes, splitting each group into multiple subgroups. Then, the refined groups are aggregated to construct a group-level graph structure, as shown in formula (2).

$$G' = \left(\bigcup_{c \in P_{\text{refined}}} C, E' \right) \tag{2}$$

where P_{refined} is the set of refined groups and E' is the set of edges between groups. Nodes G' in the graph now represent each refined group, and edges indicate relationships between groups. This group-level graph structure enables feature learning at different granularities. The group iterative identification process is illustrated in Figure 2, where Group 1 is split into two subgroups after internal node refinement and then aggregated into four groups through node aggregation. Finally, the node movement, group refinement, and aggregation processes are

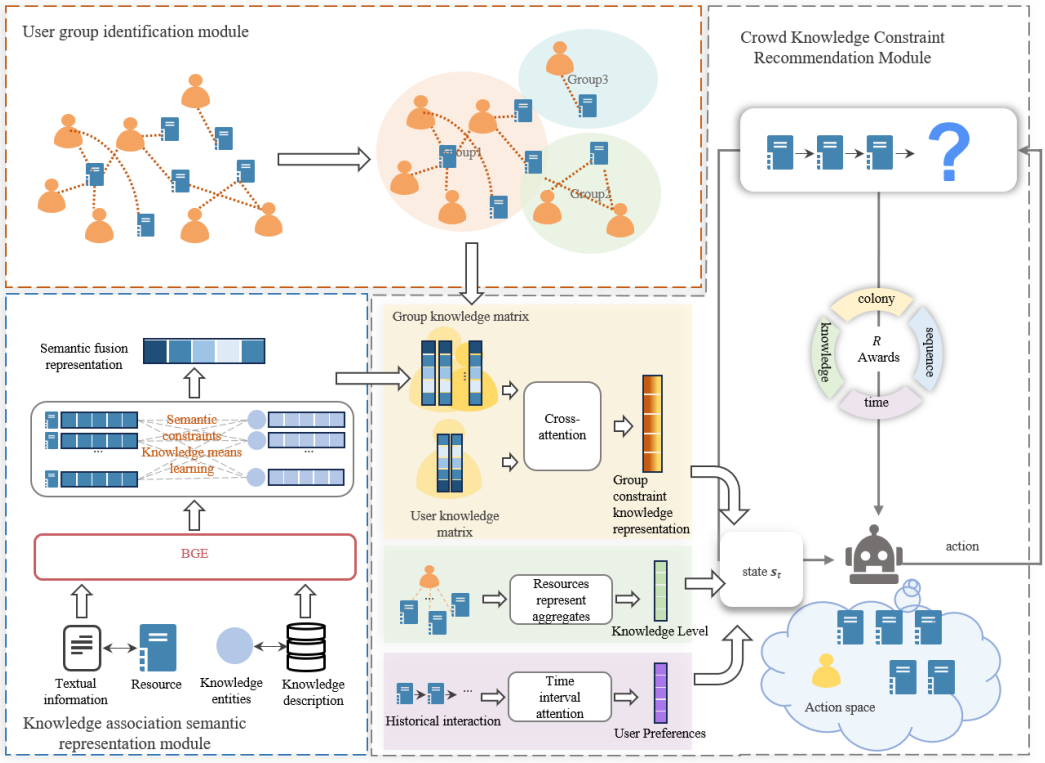


Fig. 1. Framework of GEPR, consisting of three core modules (user group identification, knowledge association semantic representation, and group knowledge-constrained recommendation). User-resource interaction data undergoes iterative user group division based on behavioral features, extracts deep features through knowledge association semantic enhancement, and generates personalized educational resource recommendations by combining the RL strategy with group knowledge constraints.

iteratively performed until the quality function reaches its maximum value, yielding the final group identification results.

3.2 Knowledge Association Semantic Representation Module

Semantic-based representation learning models focus on extracting semantic features of entities and relationships from text or other forms of semantic information. Such methods aim to map text information into a continuous vector space, where texts with similar semantics are mapped to adjacent positions, resulting in small distances between semantically similar texts in this space.With the development of deep learning, pre-trained language models represented by BERT have gradually become mainstream. Through bidirectional encoding, pre-training, and fine-tuning strategies, BERT has revolutionarily improved the capability of natural language processing, especially demonstrating excellent performance in context understanding and task adaptability.

With the growing demand for knowledge representation learning in recommendation systems, models capable of providing higher-quality and more general text representations have gradually emerged. Among them, the BGE (BAAI General Embeddings) (Xiao et al. 2024) model developed by Beijing Academy of Artificial Intelligence

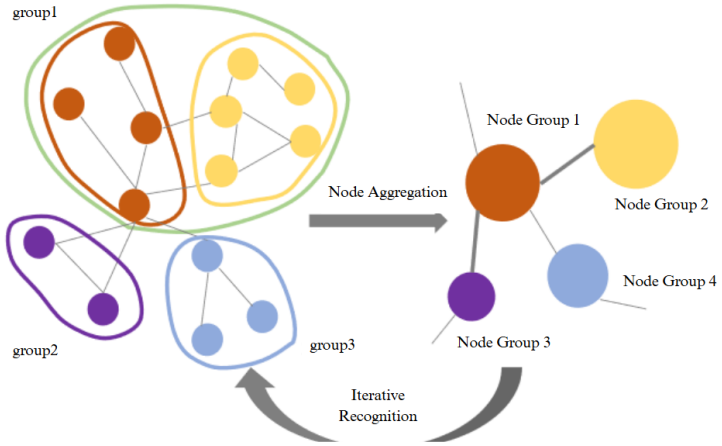


Fig. 2. Group iterative grouping process, illustrating the complete process of node movement, group refinement,correction,and aggregation.

(BAAI) has attracted significant attention due to its excellent performance and broad application prospects. BGE has distinct advantages in multilingual support, computational efficiency, and representation quality. To further improve the model's performance in semantic representation tasks, BGE has made targeted optimizations to the classic Transformer architecture, constructing a Transformer encoder-decoder framework. Specifically, the model structure of BGE consists of 12 Transformer encoders and one Transformer decoder. Each encoder layer is composed of a multi-head self-attention mechanism and a feed-forward neural network.To effectively capture semantic associations between resources and knowledge points, each resource entity and knowledge entity's text information is first encoded into vector representations using the BGE model, constructing resource and knowledge entity text semantic matrices, as shown in formula (3).

$$\begin{aligned} \boldsymbol{O}_i &= (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \text{BGE}\,(i_1, i_2, \ldots, i_n) \\ \boldsymbol{O}_k &= (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m) = \text{BGE}\,(k_1, k_2, \ldots, k_m) \end{aligned} \tag{3}$$

Second, considering that in actual educational scenarios, educational resources often cover multiple knowledge points, and even if resources share the same theme, their contents may focus on different knowledge concepts. Therefore, to effectively capture the diverse associations between educational resources and knowledge points, the knowledge associations between them are quantified through text semantic similarity to measure the association weights between resource entities and knowledge entities. The knowledge association weight matrix $W_{i,k} \in \mathbb{R}^{n \times m}$ is obtained via cosine similarity, as shown in formula (4).

$$\boldsymbol{W}_{i,k} = \text{sim}\,(\boldsymbol{O}_i, \boldsymbol{O}_k) = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix} \tag{4}$$

where w_{nm} represents the knowledge association weight between the n-th resource entity and the m-th knowledge entity. The entity representation vector is then constructed by integrating structural and semantic features, as shown in formula (5).

$$\boldsymbol{e} = \boldsymbol{h}_s \oplus \boldsymbol{h}_d \tag{5}$$

where h_s and h_d are the entity structural representations obtained via TransE, and the entity descriptive information representations obtained via BGE, respectively.

To further optimize the knowledge graph representation learning process, a semantic association constraint term is designed. This term imposes distance constraints between semantically associated entities, as shown in formula (6).

$$L_r = \sum_{(e_i,e_j)\in S_r} \max\left(0, \|e_i - e_j\|_\lambda - w_{ij}\right) \tag{6}$$

where S_r is the set of entity pairs with knowledge associations, including resource and knowledge entities, and δ is the distance norm. When the knowledge association weight between two entities is greater, it is hoped that the entities are closer in the embedding space, and the constraint term has a smaller impact on the loss function.

Finally, the loss function is defined as shown in formula (7). The training objective of the knowledge association semantic representation module is to minimize the loss function.

$$\text{Loss} = \sum_{(h,r,t)\in G} \sum_{(h',r,t')\in G'} [\gamma + f_r(e_h, e_t) - f_r(e_{h'}, e_{t'})] + \alpha L_r \tag{7}$$

where $f_r(e_h, e_t)$ is the scoring function of TransE, e_h and e_t are the embedding representations of the head and tail entities respectively. γ is the margin, and α is the influence factor that adjusts the effect of the constraint term on the embedding calculation.

After the knowledge-related semantic representation module's training, we get a semantic-enhanced resource representation matrix $F \in \mathbb{R}^{n\times d}$. For each user, we filter resources from F that are related to them, forming a group knowledge matrix F^{group} and a user knowledge matrix F^{user}. This process completes the integration of different knowledge points for all resources in the recommendation system. The semantically fused resource representations can capture the resources' inherent properties and their deep seated relations with knowledge points, providing high-quality input for personalized recommendations.

3.3 Group Knowledge Constrained Recommendation Module

To introduce broader knowledge information, the user groups divided by the interaction behavior graph are combined with the semantically fused resource representations learned through knowledge association semantic representation.

3.3.1 Group Constrained Knowledge Representation. Based on the cross-attention design concept, the association between user knowledge and group knowledge is learned. Specifically, for each user's knowledge representation learning, the group knowledge matrix serves as the query matrix, and the user knowledge matrix serves as the key and value matrices, as shown in formulas (8).

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{F}^{\text{group}} \boldsymbol{W}^{Q} \\ \boldsymbol{K} &= \boldsymbol{F}^{\text{user}} \boldsymbol{W}^{K} \\ \boldsymbol{V} &= \boldsymbol{F}^{\text{user}} \boldsymbol{W}^{V'} \end{aligned} \tag{8}$$

where $F^{\text{group}} \in \mathbb{R}^{n\times d}$ is the group knowledge matrix, formed by stacking the knowledge representations of resources within the group, representing the global knowledge of the group. $F^{\text{user}} \in \mathbb{R}^{m\times d}$ is the user knowledge matrix, formed by stacking the knowledge representations of resources in the user's interaction sequence, representing the local knowledge information of the user. W^Q, W^K, W^V are trainable weight matrices.

Then, softmax is used to normalize the attention weights, obtaining the weight of each query for different keys. The attention weight indicates the degree of focus each user has on different resources within the group

knowledge context. Next, the group constrained knowledge representation is obtained by weighting and summing the value matrix with the attention weights, as shown in formula (9).

$$g = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{9}$$

Through the cross-attention mechanism, group knowledge helps compensate for specific knowledge deficiencies of users. Each user's knowledge representation is adjusted within the group context, enriching the internal knowledge features and enhancing the diversity of recommended resource predictions.

3.3.2 LLM Enhanced Resource Representation. Knowledge graphs can provide deeper association information for more accurate user representation learning. This module samples candidate entity pairs based on the text similarity of knowledge entities. Using the powerful semantic learning ability of LLMs, it captures implicit knowledge associations between entities, optimizes candidate entity pairs, supplements triples, and enhances knowledge graph structure to obtain richer resource representations. Subsequently, the knowledge-enhanced resource representations are used to strengthen the implicit knowledge representations in user interaction sequences, providing user knowledge dimension information for educational resource recommendations. The LLM-enhanced resource representation process is shown in Figure 3. To optimize knowledge associations in

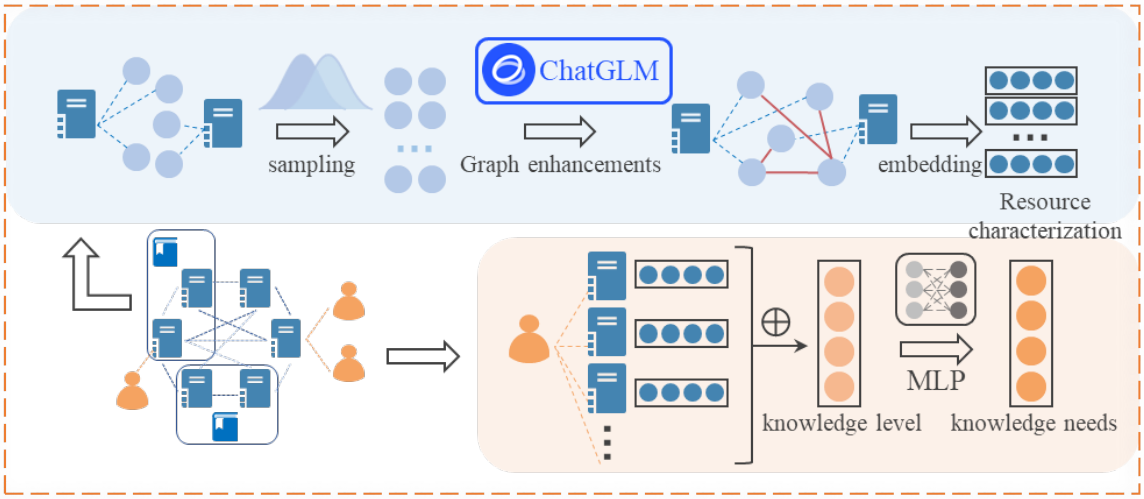


Fig. 3. LLM - enhanced resource representation, optimizing resource semantic representations through candidate entity pair filtering and knowledge graph supplementation.

educational knowledge graphs, candidate entity pairs are first screened based on semantic cosine similarity, as shown in formulas (10).

$$\begin{aligned} C(e) &= \{\, \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle \mid \text{sim}\left(\boldsymbol{e}_i, \boldsymbol{e}_j\right) \geq p, \boldsymbol{e} \in E' \} \\ \text{sim}\left(\boldsymbol{e}_i, \boldsymbol{e}_j\right) &= \frac{\boldsymbol{e}_i \cdot \boldsymbol{e}_j}{\|\boldsymbol{e}_i\| \left\|\boldsymbol{e}_j\right\|} \end{aligned} \tag{10}$$

where $C(e)$ is the set of candidate entity pairs screened based on similarity, e is the entity representation, $E' \subset E$ is the set of all knowledge hierarchy entities, and p is the predefined similarity threshold.

To fully utilize the attribute information of entities in the knowledge graph to precisely obtain the semantic associations between candidate entity pairs, LLMs are used to optimize the candidate entity pair set, as shown in formula (11).

$$\hat{y}_{ij} = \text{LLM}(P_{\langle e_i, e_j \rangle}) \tag{11}$$

where $\hat{y}_{ij}$ represents the relevance of the candidate entity pair $\langle e_i, e_j \rangle$, with a value in $\{0, 1\}$. It takes the value 1 if there is a correlation between the entities, and 0 otherwise. $P\langle e_i, e_j \rangle$ is a pre-designed prompt template, as shown in Figure 4. The prompt template encompasses various key components. The course information section includes both the course title and a detailed synopsis, enabling the large language model (LLM) to quickly grasp the core content of the course. The knowledge base section provides in - depth and meticulous explanations of relevant professional knowledge, furnishing the model with rich materials to understand the knowledge background. The example section, through specific input - output instances, vividly and clearly demonstrates how to judge the relevance of entity pairs, which is beneficial for the model to learn and imitate correct judgment logic.

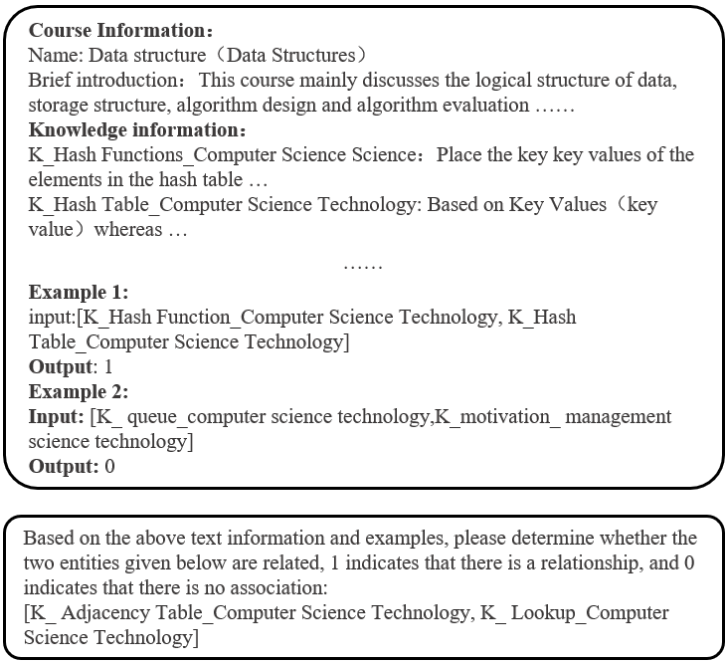


Fig. 4. Example of prompt template for ChatGLM to filter candidate entity pairs, including input format, judgment logic, and output specifications.

The text attribute information of resource entities and knowledge entities in the knowledge graph serves as the information basis for model decision - making. This information offers abundant reference content for the model from different dimensions, assisting the LLM in comprehending the current task more accurately and profoundly. Moreover, examples designed in the prompt template are highly representative and instructive. They are used to standardize the model's output, ensuring that the model can follow a unified and reasonable standard when handling similar tasks, thereby improving the consistency and reliability of the output results.

After the semantic enhancement and filtering, we get a set of knowledge entity pairs $C'(e) = \{\langle e_i, e_j\rangle \mid \hat{y}_{ij} = 1 \cap \text{sim}(e_i, e_j) \geq p\}$. We define the relationships between entities in $C'(e)$ as r_{sim}, create new knowledge triples $(e_i, r_{\text{sim}}, e_j)$, and add them to the educational knowledge graph G to form an enhanced graph G'. Then, we use TransE to obtain new resource representations e'.

Then, to better grasp and pinpoint users' knowledge needs, for user u and their interaction sequence s_u, we aggregate representations of historically interacted resources to gauge the user's current knowledge level, as shown in formula (12).

$$x_t = \text{agg}(e'_{i_1}, e'_{i_2}, \ldots, e'_{i_k}) \tag{12}$$

where x_t is the user's knowledge level at time t, e'_{i_k} is the representation of the k-th resource in the user's interaction sequence, and $\text{agg}(\cdot)$ is the aggregation function. In this paper, we use the mean of interacted resource representations to indicate the user's knowledge level.

3.3.3 User Preference Representation Based on Time Intervals. Different users may have significantly different interaction patterns within a teaching cycle. Frequent interactions may indicate strong user interest in the current content, while longer intervals may suggest changes in user preferences. To better capture dynamic changes in user preferences and enhance the recommendation system's understanding and prediction of user preferences, this study incorporates time information into the recommendation task and constructs a user preference representation module based on time intervals, as shown in Figure 5.

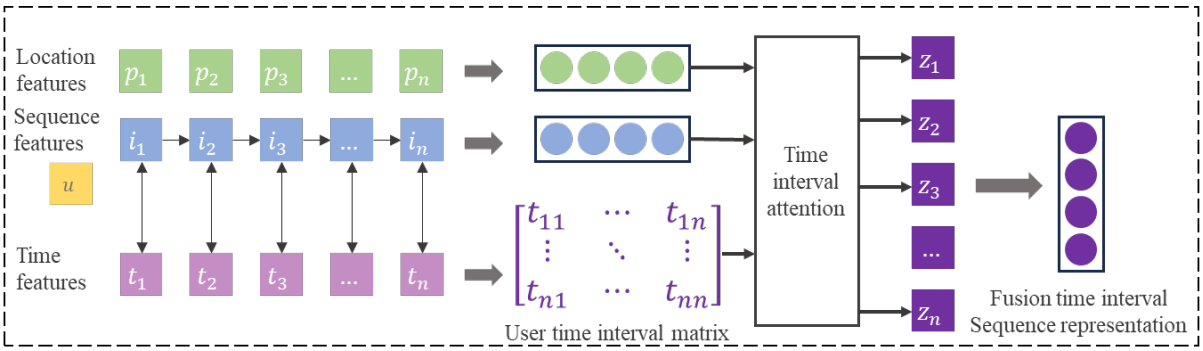


Fig. 5. User preference embedding module via time interval attention with location features, sequence features, and time interval feature fusion.

Historical Interaction Sequence Embedding. Given each user and their interaction sequence, the user's interaction resource sequence and interaction time sequence of length l are obtained. To combine the semantic information of recommended resources for user preference modeling, the enhanced resource representations from the knowledge graph are used to construct the interaction resource embedding matrix, as shown in formula (13).

$$M_{\text{emb}} = \begin{bmatrix} m_{i_1} \\ m_{i_2} \\ \vdots \\ m_{i_l} \end{bmatrix} \tag{13}$$

where m_{i_k} represents the embedding of the k-th resource interacted by the user. It is formed by concatenating the index of resource i_k representation e_k^{id} and the graph-enhanced representation e'_k, as shown in $m_{i_k} = \text{concat}(e_k^{\text{id}}, e'_k)$.

In the historical interaction sequence, user interest is reflected not only in the features of the resources they interact with but also in the order of interaction. To preserve the model's ability to sense the order of user-resource interactions, a position embedding matrix M_{pos} is introduced, as shown in formula (14).

$$M_{\text{pos}} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_l \end{bmatrix} \tag{14}$$

where p_l denotes the position embedding of the l-th resource in the interaction sequence.

Inspired by TiSASRec , which introduced a time interval-aware self-attention mechanism for sequential recommendation tasks. It incorporates time interval information into self-attention to better capture the dynamic changes of user interest, enhancing recommendation system performance. As different users have varying interaction time patterns, when modeling user preferences, only the relative time interval information of individual users is considered. We construct an interaction time interval matrix M_{time} for user u, as shown in formula (15).

$$M_{\text{time}} = \begin{bmatrix} t_{11} & \cdots & t_{1l} \\ \vdots & \ddots & \vdots \\ t_{l1} & \cdots & t_{ll} \end{bmatrix} \tag{15}$$

where t_{ij} denotes the time interval between when a user accesses resources i and j. To prevent excessive intervals from causing sparsity in the time interval matrix, a maximum threshold $t_{\max}$ is set, ensuring $t_{ij} = \min(t_{\max}, t_{ij})$. By constructing a time interval matrix for each user, the model can more accurately track the dynamic changes in user behavior.

Time Interval Attention Mechanism. For each user u's interaction sequence s_u, we generate a sequence representation $Z = (z_1, z_2, \ldots, z_l)$ of length l that incorporates time interval information. We define the linear transformation matrices W^Q, W^K, and W^V to obtain the Q, K, and V matrices in the self-attention mechanism, as shown in formulas (16)-(18). To enhance the model's ability to perceive multiple features of the interaction sequence, we add positional and time interval information to the K matrix.

$$Q = M_{\text{emb}} W^Q \tag{16}$$

$$K = M_{\text{emb}} W^K + P^K + T^K \tag{17}$$

$$V = M_{\text{emb}} W^V \tag{18}$$

where P^K and T^K are obtained by performing linear transformations on the position embedding matrix M_{pos} and the interaction time interval matrix M_{time}, respectively.

The dependency relationships between resources in the interaction sequence are quantified by calculating the attention scores between resources i and j, as shown in formula (19).

$$\text{score}_{ij} = \frac{Q_i \cdot K_j^T}{\sqrt{d}} \tag{19}$$

where d is the vector dimension used for scaling the attention scores.

The attention weights are normalized using the softmax function, as shown in formula (20).

$$\alpha_{ij} = \text{softmax}(\text{score}_{ij}) \tag{20}$$

Finally, the sequence representation incorporating time interval information is calculated based on the attention weights, as shown in formula (21).

$$z_i = \sum_{j=1}^{l} \alpha_{ij} V_j \tag{21}$$

To further model user preference representations, the sequence representation incorporating time interval information is aggregated, as shown in formula (22).

$$h = \text{agg}(z_1, z_2, \ldots, z_l) \tag{22}$$

where h is the final user preference representation, and $\text{agg}(\cdot)$ is an aggregation function calculated via average pooling.

In the time interval attention mechanism, the model can more flexibly focus on the impacts of recent or long-term interactions based on user behavior, effectively capturing trends in user preference changes. Additionally, this module fully utilizes the inherent characteristics of resources, providing a refined and dynamic approach to user behavior modeling.

3.3.4 Joint Reward Design. The personalized recommendation process of users is modeled as a Markov decision process. The group-constrained knowledge representation, user knowledge level, and user preference representation obtained from time interval attention are combined to form the state representation of the RL agent, as shown in formula (23).

$$s_t = \text{concat}(g_t, x_t, h_t) \tag{23}$$

where g_t, x_t, and h_t are the group-constrained knowledge representation, user knowledge level, and user preference representation of the current user at time t, obtained from formulas (9), (12), and (22), respectively.

Joint Reward. To provide richer feedback information to the agent and improve learning efficiency and adaptability to complex recommendation tasks, a joint reward function incorporating knowledge, sequence, time, and group constraints is designed, as shown in formula (24).

$$R = R_{\text{kg}} + R_{\text{seq}} + R_{\text{time}} + \beta R_{\text{group}} \tag{24}$$

where R_{kg} is the knowledge reward, R_{seq} is the sequence reward measuring the difference between predicted and actual recommendation sequences, R_{time} is the time reward calculated by measuring the knowledge similarity between user preference representations and actual recommendation results, and R_{group} is the group constraint reward. β is a weight factor controlling the influence of group constraints on the joint reward and, consequently, on action selection.

The knowledge reward is defined as the cosine similarity between the predicted user knowledge level and the actual knowledge level, as shown in formula (25).

$$R_{\text{kg}} = \text{sim}(\hat{x}_{t+k}, x_{t+k}) \tag{25}$$

The BLEU (Papineni et al. 2002) (Bilingual Evaluation Understudy) metric, commonly used in machine translation evaluation, assesses translation quality by comparing machine-translated text with multiple reference translations. It calculates the precision of n-gram (sequences of n words) matches between them.

If the predicted sequence $\hat{i}_{t:t+k} = (\hat{i}_t, \hat{i}_{t+1}, \ldots, \hat{i}_{t+k})$, each resource $\hat{i}_t$ in the sequence is treated as a "word", and the entire sequence as a "sentence". The difference between the predicted and actual sequences is used to evaluate the recommendation sequence's overall performance, as shown in formula (26).

$$R_{\text{seq}} = \exp\left(\sum_{j=1}^{N} \omega_n \log \text{prec}_n\right) \tag{26}$$

where N is the maximum length used in the evaluation, ω_n is the weight of n-gram, calculated as $1/N$; prec_n represents the n-gram similarity between the predicted and actual sequences.

The time reward is calculated as the cosine similarity between the user preference representation and the actual recommendation sequence's user knowledge level, as shown in formula (27).

$$R_{\text{time}} = \text{sim}(h_{t+k}, x_{t+k}) \tag{27}$$

The group constraint reward is shown in formula (28).

$$R_{\text{group}} = \text{avg}\left(\sum_{t}^{t+k} \text{sim}(g_t, q_{a_t})\right) \tag{28}$$

where g_t is the group-constrained knowledge representation, q_{a_t} is the action representation, and the similarity function $\text{sim}(\cdot)$ employs cosine similarity. The cumulative mean from the current timestep t to the future timestep $t + k$ is calculated to serve as a smoothed reward.

Policy Network. In RL, the policy network's core goal is to learn a mapping probability distribution function between states and actions, enabling the agent to select optimal strategies in specified states to maximize cumulative rewards. In sequence recommendation scenarios, user feedback data is sparse, and Monte Carlo policy methods are suitable for sparse reward environments as they estimate gradients based on complete trajectories rather than relying solely on immediate rewards. Complete trajectories in sequence recommendation encompass multiple user-resource interactions, providing a comprehensive reflection of user preferences and behavior patterns.

Specifically, for each state s_t, the policy $\pi(a_t|s_t)$ is defined as follows, calculating the probability of the agent executing all possible actions in the current state, as shown in formula (29).

$$\pi(a_t|s_t) = \frac{\exp\{q_{i_j(a_t)} W s_t\}}{\sum_{i_j \in I} \exp\{q_{i_j} W s_t\}} \tag{29}$$

where q_{i_j} denotes the embedding of the j-th resource. $q_{i_j(a_t)}$ denotes the embedding of the resource corresponding to action a_t taken by the agent. W is a parameter matrix. Guided by its policy, the agent takes an action a_t, selecting a resource from the resource set I to recommend.

The agent interacts with the environment via policy π, generating trajectories. For each timestep in a trajectory, cumulative rewards from that timestep onward are calculated. The policy network's training goal is to maximize the expected cumulative reward $J(\theta)$, defined as shown in formula (30).

$$J(\theta) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{u} \sum_{t=0}^{T} \gamma^t r_t\right] \tag{30}$$

where τ is the trajectory generated by policy π, γ is the discount factor balancing the importance of rewards, and r_t is the reward at time step t. The gradient of $J(\theta)$ is shown in formula (31).

$$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{u}\sum_{t=0}^{T}\gamma^t r_t \nabla \log \pi(a_t|s_t;\theta)\right] \tag{31}$$

where θ denotes all learnable parameters. Policy parameters are updated via gradient ascent. To reduce computational scale and parameters, trajectory truncation is used. Specifically, cumulative rewards are calculated only from the current time step t to future time step $t + k$.

4 Experiments

To evaluate the effectiveness of GEPR in educational resource recommendation tasks, experiments are conducted on two public online education datasets: MOOCCube and MOOPer. The basic information of the datasets is shown in Table 1.

4.1 Datasets and Evaluation Metrics

Table 1. Statistics information of datasets

Metrics	MOOCCube	MOOPer
Number of users	48640	42392
Triples	1743369	81120
Entity types	8	11
Relation types	12	13
Interaction records	4663919	2532524
Resources	38181 (Video)	4550 (Challenge)

MOOCCube(http://moocdata.cn/data/MOOCCube) provides detailed data on courses, teaching videos, knowledge concepts, MOOC user course selections, and video interactions, along with related knowledge graphs. MOOPer(https://www.educoder.net/ch/rest) primarily offers interaction data and knowledge graphs. Interaction data includes millions of interaction records from platform users participating in practice exercises between 2018 and 2019. The knowledge graph is constructed from entity attribute information and relationships such as courses, practices, levels, and knowledge points. MOOCCube is an open data repository for researchers in natural language processing, knowledge graphs, data mining, and other fields related to large-scale online education. It includes 706 real online courses, 38,181 teaching videos, 114,563 concepts, hundreds of thousands of course enrollment and video viewing records of 199,199 MOOC users, a concept graph with relationships such as prerequisite and hyponym between concepts, and a supplementary resource library containing hundreds of thousands of academic papers related to in-course concepts.

MOOPer is mainly divided into two parts: interaction data and knowledge graph. The interaction data includes 2,532,524 practical exercise records, 21,606,390 system feedback records, and 15,054 user forum discussion records. The knowledge graph contains 11 types of entities and 10 types of relations. The interaction data includes millions of interaction records of platform users participating in practical exercises from 2018 to 2019, and the knowledge graph is constructed from the attribute information and mutual relationships of entities such as courses, practices, levels, and knowledge points. Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) are used as evaluation metrics. The HR@K metric measures whether the model's top K predicted recommendations include user-interested resources, as shown in formula (32).

$$\text{HR@K} = \frac{1}{n}\sum_{i=1}^{n}\text{hit}(i) \tag{32}$$

where n is the total number of test samples, and $\text{hit}(i)$ is an indicator function that takes the value 1 when the user's interested resource is included in the top K recommendations and 0 otherwise.

The NDCG metric considers both whether the recommended resources are hit and their ranking positions in the recommendation list, thereby more comprehensively measuring the quality of the recommendation list. It assesses whether the recommendation system can rank user-interested resources higher in the list, as shown in formula (33).

$$\text{NDCG} = \frac{1}{n}\sum_{i=1}^{k}\frac{1}{\log_2(i+1)} \tag{33}$$

where n is the total number of test samples, and k denotes the first k items in the recommendation list. A higher user-interested resource ranked closer to the top of the list results in a higher NDCG score.

4.2 Experimental Setup

The experiments were conducted on an NVIDIA A30 24GB graphics card, using PyTorch version 2.1.1. The parameter settings required for model training include: the similarity threshold p for candidate triple filtering is 0.8; the dimensions of pre-trained knowledge graph embedding representations, user knowledge level representations, and sequence representations are all set to 50; the lengths of interaction sequences intercepted from both the MOOCCube and MOOPer datasets are set to $L = 50$; the training batch size is 2048, and the Adam algorithm is used to optimize all trainable parameters.

In the user group identification module based on behavioral characteristics, the resolution parameters on the MOOCCube and MOOPer datasets are set to $\mu = 1.5$ and $\mu = 1.0$, respectively. In the knowledge-associated semantic representation module, text descriptions (the "text" attribute and "explanation" attribute) from 38181 resource entities "video" and 114563 knowledge entities "concept" in the MOOCCube dataset, as well as texts from "topics" (3277), "discipline" (13), "subdiscipline" (58), and "challenge" (4550) entities in the MOOPer dataset, are semantically embedded based on BGE. Following most knowledge representation learning models, the set of margin γ values is $\{1.0, 2.0\}$. Through experiments, the margin γ is set to 1.0 on both datasets, the constraint term influence factor $\alpha = 0.5$, the L_1 norm is used to calculate the scoring function, and the number of training iterations is set to 1000; in the group knowledge constraint recommendation module, the ranges of the group reward weight factor β on the MOOCCube and MOOPer datasets are set to $[0.6, 0.8]$ and $[0.4, 0.6]$, respectively. To screen valid triples with genuine pedagogical logical associations in the educational knowledge graph, this study adopts a three-stage process: firstly, text normalization preprocessing is performed on resource entities (MOOCCube's teaching videos/courses, MOOPer's practical challenges) and knowledge point entities, including cleaning special symbols, redundant spaces, and meaningless filler text, as well as supplementing core definition information for short-text entities with only names but no descriptions; subsequently, the BGE-base-zh-v1.5 model is used to generate semantic embeddings for all normalized entities, with the input length adapted to the requirements of entity text, batch processing employed to balance generation efficiency and GPU memory usage, and embedding vectors normalized to ensure the accuracy of subsequent similarity calculations; then, cosine similarity is utilized to quantify the degree of semantic association between the embeddings of resource entities and knowledge point entities, with a threshold set at 0.8 (verified to cover 92% of truly associated entity pairs while filtering out 75% of redundant ones), and candidate triples are selected from entity pairs meeting the similarity threshold (1.26 million for the MOOCCube dataset and 28,000 for the MOOPer dataset); finally, a locally deployed ChatGLM3-6B model conducts binary classification on the candidate triples to eliminate redundant data

that only overlaps in keywords but lacks actual pedagogical logical associations (deemed valid if the resource content directly focuses on teaching the knowledge point). The sampling temperature is adjusted to reduce output randomness, nucleus sampling is adopted to focus on high-probability semantic expressions, and the output length is limited to ensure only binary results (0 for invalid association, 1 for valid association). Ultimately, 187,000 valid triples are retained for the MOOCCube dataset and 5,000 for the MOOPer dataset. Manual annotation of 1,000 randomly selected filtered triples yields an accuracy of 92.3%, and all valid triples are integrated into the original knowledge graph to complete the optimization of the graph structure.

To fully utilize the text information within the datasets, text information from course entities and concept entities in the MOOCCube dataset is selected as the text content in the prompt template; in the MOOPer dataset, text information from "topics" (3277), "discipline" (13), "subdiscipline" (58), and "challenge" (4550) entities is used to construct prompts. Dynamic division based on the timestamps of user interaction sequences: each user's interaction sequence is sorted by time, with the first 80% used as the training set, the middle 10% as the validation set, and the last 10% as the test set, ensuring that each user has at least 1 test sample. Users with fewer than 3 interactions are filtered out during the division process, and finally, 48,638 users are retained in MOOCCube and 42,390 users in MOOPer.The "videos" in MOOCCube are instructional videos covering 8 subject areas, accompanied by knowledge point annotations and textual descriptions. The "Challenges" in MOOPer are collections of multi-level practical exercises focusing on single knowledge points, where each Challenge consists of 3-10 levels and provides real-time feedback (e.g., answer accuracy rate, error analysis). After filtering by the large language model, a total of 890356 knowledge triples are added to the MOOCCube dataset, accounting for 31.6% of the KG triples after data preprocessing; in the MOOPer dataset, a total of 20273 knowledge triples are added, accounting for 20.15% of the KG triples after data preprocessing.

The experimental hardware environment consists of NVIDIA A30 24GB graphics cards. The time breakdown of a complete experimental cycle (from raw data to final evaluation results) is as follows: Data preprocessing phase: including text cleaning, BGE embedding generation, and ChatGLM-based triple filtering, taking approximately 75 minutes for the MOOCCube dataset and 25 minutes for the MOOPer dataset; Model training phase: an early stopping strategy is adopted—user group identification takes about 12 minutes (MOOCCube) and 8 minutes (MOOPer), while the training of the RL-based recommendation strategy takes approximately 105 hours (MOOCCube, 140 epochs, 45 minutes per epoch) and 60 hours (MOOPer, 120 epochs, 30 minutes per epoch); Model evaluation phase: full-scale evaluation based on multi-dimensional Top-K metrics (K=5/10/20) takes about 30 minutes (MOOCCube) and 20 minutes (MOOPer).

In summary, a complete experimental cycle takes approximately 108 hours for the MOOCCube dataset and 61 hours for the MOOPer dataset. With multi-GPU parallel training (e.g., 4 NVIDIA A30 graphics cards), the training phase can be reduced to 38 hours (MOOCCube) and 25 hours (MOOPer), and the overall experimental time can be compressed to 26 hours (MOOCCube) and 15 hours (MOOPer), demonstrating time feasibility for practical engineering deployment.

4.3 Parameter Sensitivity Experiments

To explore the impact of group size on the effectiveness of educational resource recommendation, experiments were conducted on both MOOCCube and MOOPer datasets.

4.3.1 Impact of Group Number on Educational Resource Recommendation. A set of resolution parameters, specifically $u = \{0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$, was selected for investigation. The number of group divisions under different resolution settings, along with the corresponding experimental results, are detailed in Tables 2 and 3. In order to more intuitively observe how resolution affects group partitioning, the group divisions under various resolution parameters have been visualized, as presented in Figures 6 and 7. All experimental settings are run five

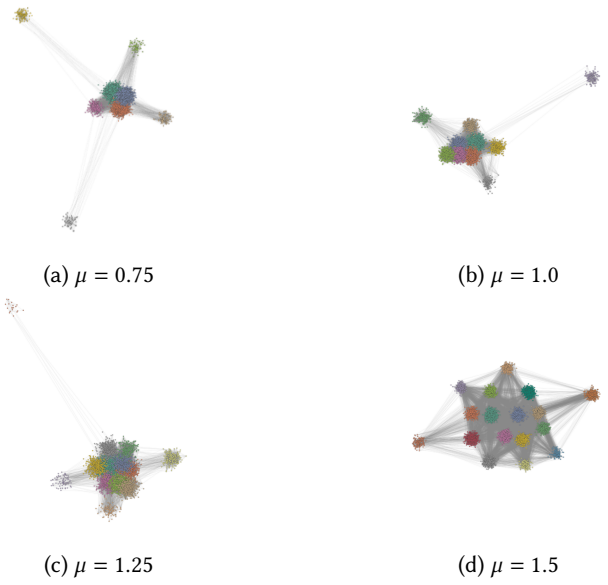
times independently; the averaged results and standard deviations are presented as mean±std, with statistical comparisons conducted using the paired t-test at the 0.01 significance level.

Table 2. Comparison results under different μ for the MOOCCube

μ	Groups	HR@1	HR@3	HR@5	HR@10	NDCG@10
0.75	8	0.8130±0.0011	0.9083±0.0007	0.9284±0.0015	0.9432±0.0003	0.8794±0.0017
1.0	10	0.8176±0.0005	0.9097±0.0018	0.9277±0.0009	0.9446±0.0012	0.8789±0.0004
1.25	13	0.8180±0.0014	0.9109±0.0006	0.9285±0.0013	0.9453±0.0008	0.8815±0.0016
1.5	17	0.8180±0.0009	0.9117±0.0015	0.9330±0.0002	0.9482±0.0011	0.8820±0.0005
1.75	22	0.8176±0.0017	0.9104±0.0003	0.9276±0.0018	0.9447±0.0006	0.8809±0.0012
2.0	29	0.8118±0.0008	0.9087±0.0012	0.9271±0.0004	0.9407±0.0019	0.8769±0.0009

Table 3. Comparison results under different μ for the MOOPer

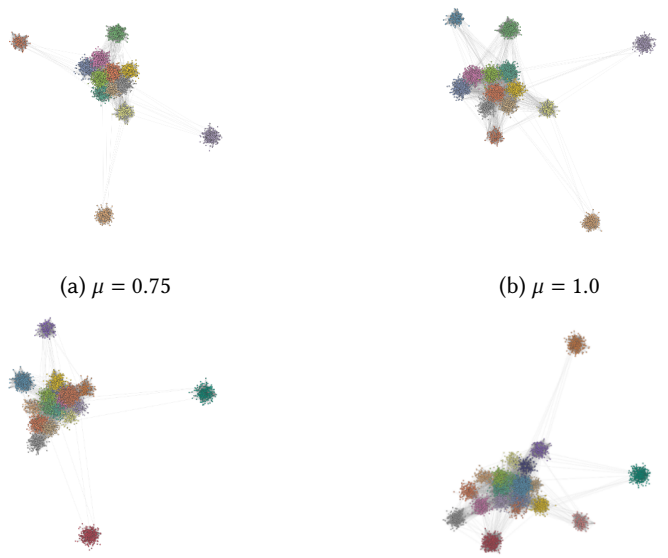
μ	Groups	HR@1	HR@3	HR@5	HR@10	NDCG@10
0.75	13	0.9270±0.0014	0.9712±0.0006	0.9841±0.0018	0.9884±0.0015	0.9629±0.0013
1.0	14	0.9279±0.0013	0.9718±0.0016	0.9845±0.0005	0.9897±0.0011	0.9636±0.0015
1.25	18	0.9265±0.0003	0.9702±0.0018	0.9839±0.0015	0.9895±0.0013	0.9634±0.0016
1.5	20	0.9261±0.0010	0.9713±0.0004	0.9842±0.0019	0.9886±0.0012	0.9637±0.0013
1.75	22	0.9247±0.0004	0.9709±0.0016	0.9834±0.0014	0.9887±0.0010	0.9624±0.0009
2.0	24	0.9253±0.0013	0.9697±0.0008	0.9837±0.0003	0.9871±0.0004	0.9628±0.0011



(a) $\mu = 0.75$ (b) $\mu = 1.0$

(c) $\mu = 1.25$ (d) $\mu = 1.5$

Fig. 6. Visualization of user group division under different resolutions on the MOOCCube dataset.

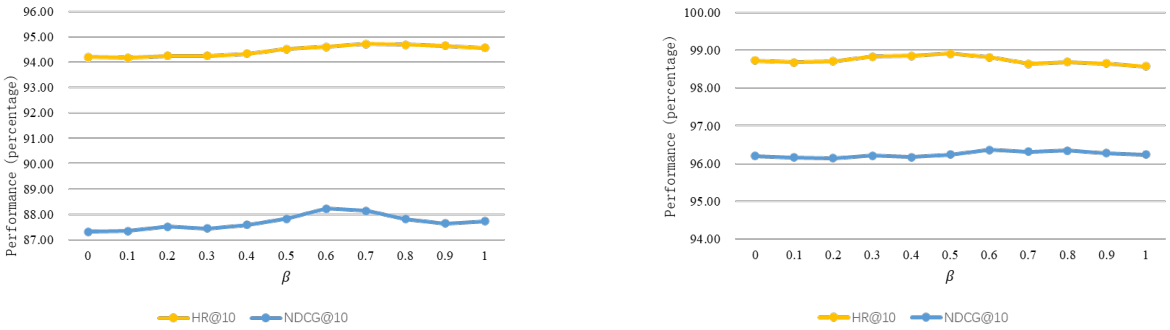


(a) $\mu = 0.75$ (b) $\mu = 1.0$

(c) $\mu = 1.25$ (d) $\mu = 1.5$

Fig. 7. Visualization of user group division under different resolutions on the MOOPer dataset.

4.3.2 Impact of Group Reward on Educational Resource Recommendation. To investigate the impact of group reward on educational resource recommendation, the weight factor of group reward is set to range from 0 to 1 with a fixed step size of 0.1. Experiments are conducted on MOOCCube and MOOPer datasets to observe changes in recommendation performance and determine the optimal weight setting range. Taking HR@10 and NDCG@10 as examples, the experimental results are shown in Figure 8.



(a) MOOCCube (b) MOOPer

Fig. 8. Effects of different group reward weight factors (β) on recommendation performance, with HR@10 and NDCG@10 as evaluation metrics.(the y-axis uses a truncated scale)

From Figure 8, on the MOOCCube and MOOPer datasets, HR and NDCG metrics first rise then fall as β increases. On MOOCCube, the model performs best when β is $0.6 - 0.8$; On MOOPer, it's $0.4 - 0.6$. Low β makes

the model focus too much on individual user sequence features, leading to insufficient signals in data-sparse scenarios and affecting recommendation accuracy. But too high β makes the agent rely overly on group features, interfering with the model's learning of user-specific features and causing recommendation results to deviate from user needs, thus reducing overall recommendation performance. In summary, introducing proper group rewards into the joint reward function can ensure the agent balances the learning of individual and group features during decision-making, achieving optimal results.

4.4 Experiments

To verify the comprehensive recommendation performance of GEPR, comparative experiments are conducted on MOOCCube and MOOPer datasets against nine recommendation methods, including sequence-based methods (GRU4Rec, SASRec, FEARec, HRL), graph-based methods (LightGCN, KGCN, KGAT), and hybrid information-based methods (KERL, MkEGC). The comparative experimental results are shown in Tables 4 and 5. Bold indicates the best results, and underlined indicates the second-best results.

Table 4. Comparison results on MOOCCube.(using paired t-test at 0.01 significance level)

Model	HR@1	HR@3	HR@5	HR@10	NDCG@10
GRU4Rec	0.5352±0.0018	0.6349±0.0023	0.6670±0.0015	0.7057±0.0021	0.6205±0.0019
SASRec	0.4648±0.0022	0.6520±0.0017	0.6953±0.0020	0.7441±0.0014	0.6106±0.0023
FEARec	0.4497±0.0019	0.6528±0.0024	0.6997±0.0016	0.7445±0.0022	0.6057±0.0017
HRL	0.4708±0.0025	0.7091±0.0018	0.7893±0.0021	0.8581±0.0013	0.5609±0.0020
LightGCN	0.4945±0.0020	0.7341±0.0015	0.8191±0.0024	0.8996±0.0018	0.4687±0.0016
KGCN	0.2109±0.0023	0.3894±0.0019	0.4893±0.0022	0.6278±0.0015	0.2184±0.0018
KGAT	0.3470±0.0021	0.5909±0.0023	0.6954±0.0017	0.8190±0.0020	0.3449±0.0015
KERL	0.7797±0.0016	0.8871±0.0020	0.9101±0.0014	0.9375±0.0023	0.8619±0.0017
MkEGC	<u>0.7931±0.0014</u>	<u>0.8920±0.0018</u>	<u>0.9185±0.0022</u>	<u>0.9463±0.0012</u>	<u>0.8731±0.0019</u>
GEPR	**0.8180±0.0011**	**0.9117±0.0017**	**0.9330±0.0009**	**0.9482±0.0012**	**0.8820±0.0015**

Table 5. Comparison results on MOOPer

Model	HR@1	HR@3	HR@5	HR@10	NDCG@10
GRU4Rec	0.7013±0.0019	0.7515±0.0022	0.7608±0.0017	0.7901±0.0024	0.7325±0.0015
SASRec	0.6771±0.0021	0.7526±0.0018	0.7730±0.0023	0.8177±0.0016	0.7350±0.0020
FEARec	0.6835±0.0023	0.7604±0.0015	0.7896±0.0020	0.8340±0.0018	0.7428±0.0022
HRL	0.6873±0.0017	0.8532±0.0024	0.8773±0.0016	0.9038±0.0021	0.7931±0.0014
LightGCN	0.6342±0.0020	0.8033±0.0019	0.9173±0.0022	0.9541±0.0015	0.6873±0.0018
KGCN	0.4221±0.0025	0.5952±0.0021	0.7463±0.0017	0.8368±0.0023	0.3913±0.0016
KGAT	0.4708±0.0022	0.6473±0.0018	0.7985±0.0024	0.8854±0.0015	0.4727±0.0020
KERL	0.9132±0.0015	0.9585±0.0020	0.9762±0.0027	0.9807±0.0022	0.9496±0.0014
MkEGC	<u>0.9148±0.0018</u>	<u>0.9612±0.0023</u>	<u>0.9769±0.0015</u>	<u>0.9893±0.0020</u>	<u>0.9562±0.0017</u>
GEPR	**0.9279±0.0016**	**0.9718±0.0021**	**0.9845±0.0018**	**0.9897±0.0014**	**0.9636±0.0020**

From the analysis of Tables 4 and 5:

GEPR achieves the best results on both datasets. Compared to the second-best method MkEGC, GEPR shows improvements of 2.49%, 1.97%, 1.45%, 0.19%, and 0.89% in HR@1, HR@3, HR@5, HR@10, and NDCG@10 metrics on the MOOCCube dataset, respectively. On the MOOPer dataset, the improvements are 1.31%, 1.06%, 0.76%, 0.04%, and 0.74%, respectively.

MkEGC, although applying dual graph convolution and hierarchical weighting strategies to optimize educational resource representation learning, focuses solely on feature aggregation from a single-user interaction perspective and underutilizes textual information of resources. GEPR enhances resource representation by incorporating semantic association learning in the textual domain and enriches user knowledge representation learning with group knowledge. This enables the model to better capture user behavior patterns and cross-user commonalities, thereby improving recommendation performance.

Compared to the MOOPer dataset, GEPR achieves greater performance improvements on the MOOCCube dataset. Analysis of the two datasets reveals that MOOCCube is sparser than MOOPer, suggesting that GEPR can effectively handle recommendation tasks in data-sparse scenarios.

4.5 Ablation Study

To validate the effectiveness of the user group identification module, knowledge association semantic representation module, and group knowledge constraint reward in GEPR, ablation experiments are conducted. Tables 6 and 7 present the ablation experimental results on the MOOCCube and MOOPer datasets. Specifically, the ablation experiments involve the following settings:

(1) w/o BGE: Textual semantic representation learning is not introduced, and entity embeddings are obtained solely via TransE.
(2) w/o L_r: The semantic similarity constraint term L_r is not introduced, and knowledge graph representation learning is optimized solely based on textual semantic representations.
(3) w/o Group: Group knowledge enhancement is not employed, and knowledge representation learning is based solely on single-user knowledge features.

Table 6. Results of ablation study on MOOCCube

Model	HR@1	HR@3	HR@5	HR@10	NDCG@10
w/o BGE	0.8108±0.0014	0.9054±0.0012	0.9255±0.0017	0.9415±0.0011	0.8787±0.0015
w/o L_r	0.8094±0.0016	0.9061±0.0013	0.9263±0.0011	0.9399±0.0018	0.8753±0.0012
w/o Group	0.8126±0.0013	0.9070±0.0015	0.9271±0.0014	0.9445±0.0012	0.8769±0.0017
GEPR	**0.8180±0.0012**	**0.9117±0.0011**	**0.9330±0.0015**	**0.9482±0.0010**	**0.8820±0.0013**

Table 7. Results of ablation study on MOOPer

Model	HR@1	HR@3	HR@5	HR@10	NDCG@10
w/o BGE	0.9237±0.0013	0.9678±0.0015	0.9812±0.0011	0.9870±0.0017	0.9605±0.0012
w/o L_r	0.9239±0.0016	0.9681±0.0012	0.9820±0.0014	0.9876±0.0011	0.9611±0.0018
w/o Group	0.9236±0.0011	0.9698±0.0017	0.9821±0.0015	0.9879±0.0013	0.9621±0.0014
GEPR	**0.9279±0.0012**	**0.9718±0.0011**	**0.9845±0.0013**	**0.9897±0.0010**	**0.9636±0.0015**

Based on the experimental results in Tables 6 and 7, Ablation Study show that after removing BGE text semantic embedding (w/o BGE), the NDCG@10 of the MOOCCube dataset decreases by 0.33%, and after removing the semantic constraint term (w/o L_r), it decreases by 0.67%, the following conclusions can be drawn:

(1) Introducing textual semantic embeddings can effectively enhance entity feature representations and improve recommendation performance. Specifically, textual semantic embeddings utilize the textual descriptions of educational resources and knowledge points to further uncover semantic associations between entities, providing additional information sources for optimizing knowledge graph representations. This enhancement is validated on both datasets. The smaller improvement observed on the MOOPer dataset may be attributed to the limited additional descriptive information for concept entities in MOOPer, primarily consisting of brief textual titles, which restricts the depth of semantic learning.
(2) Incorporating textual semantic representations and semantic similarity constraints during triple representation learning aids the model in capturing semantic associations between entities. This enhances the model's semantic understanding capabilities and results in richer entity embeddings.
(3) Group knowledge enhancement based on interaction feature-based group division and its utilization to enrich user knowledge representation learning can effectively boost recommendation performance. By incorporating group knowledge, the model can complement information during cross-attention calculations for user knowledge representations. This enriches the internal knowledge features of users and improves the accuracy of recommendations.

5 Conclusion

This paper addresses the issue of sparse user interaction data in online education platforms by proposing a Group-knowledge Enhanced Personalized educational resource Recommendation (GEPR) model. GEPR constructs a user-resource interaction behavior graph and applies community detection methods to divide user groups, laying the foundation for the utilization of group knowledge. The model designs a knowledge association semantic representation module that encodes the textual information of educational resources and knowledge points using the BGE model. This maps resources and knowledge points to the same semantic space and deeply explores the semantic associations between resources and knowledge points, thereby optimizing the knowledge graph representation learning process. Additionally, GEPR introduces a cross-attention mechanism to dynamically fuse group and user knowledge. It incorporates group knowledge into the state representation and joint reward function of RL to optimize recommendation strategies. The supplementary effect of group implicit knowledge in sparse data scenarios is significant: on the MOOCCube dataset (with sparser interaction data), GEPR achieves a 2.49% improvement in HR@1 compared to the second-best method MkEGC, which is much higher than the 1.31% improvement on the MOOPer dataset. This proves that in user groups divided by the Leiden algorithm, cross-user common knowledge can effectively make up for the lack of individual user behavior data and solve the problem of insufficient personalized feature mining.

Our study only covers two educational scenarios (MOOC video learning, practical challenges) (excluding K12 vocational education), and dataset interaction behaviors are limited to "video watching, challenge completion" (lacking modeling of multi-dimensional behaviors like collecting or note-taking). The iterative group division and LLM-based knowledge graph enhancement increase computational overhead, the group division resolution parameterμ requires manual adjustment across datasets (no adaptive mechanism for different data distributions). The core framework of GEPR can be migrated to other educational scenarios, such as exercise recommendation in K12 education and skill course recommendation in vocational education . The model performs optimally on two datasets with different degrees of data sparsity, indicating strong adaptability to the sparsity of interaction data. However, in extreme cold-start scenarios , the initialization of group knowledge still needs to rely on prior information such as resource labels, resulting in limited generalization ability. Future research can conduct

in-depth exploration on the joint modeling of users' multi-behavior sequences, and leverage multi-task learning mechanisms to mine the synergistic information among sequences, thereby achieving a more comprehensive modeling of user behavior features. Meanwhile, it is worth investigating adaptive group partitioning algorithms to enhance the model's adaptability to diverse data distributions. In addition, lightweight alternatives to large language models (LLMs) can be explored to cut down model deployment costs and further expand its application scenarios.

Acknowledgments

This work is supported by the National Natural Science Foundation of china under Grant 62402160, the Natural Science Foundation of Hebei Province under Grant F2024202078.

References

Q. Ban, W. Wu, W. Hu, H. Lin, W. Zheng, and L. He. 2022. "Knowledge-enhanced multi-task learning for course recommendation." In: *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer-Verlag, Cham, Switzerland, 85–101.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. "Translating embeddings for modeling multi-relational data." In: *Advances in Neural Information Processing Systems 26*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. "Learning phrase representations using RNN encoder–decoder for statistical machine translation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, Stroudsburg, PA, USA, 1724–1734.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. "BERT: Pre-training of deep bidirectional transformers for language understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, Minneapolis, MN, USA, 4171–4186.

Y. Dong, Y. Liu, Y. Dong, Y. Wang, and M. Chen. 2024. "Multi-knowledge enhanced graph convolution for learning resource recommendation." *Knowledge-Based Systems*, 291, 111521.

X. Du, H. Yuan, P. Zhao, J. Qu, F. Zhuang, G. Liu, Y. Liu, and V. S. Sheng. 2023. "Frequency enhanced hybrid attention network for sequential recommendation." In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, USA, 78–88.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Dec. 1992. "Using collaborative filtering to weave an information tapestry." *Commun. ACM*, 35, 12, (Dec. 1992), 61–70.

J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang, and P. S. Yu. 2020. "Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view." In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, USA, 79–88.

X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. 2020. "LightGCN: Simplifying and powering graph convolution network for recommendation." In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, USA, 639–648.

B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. 2016. "Session-based recommendations with recurrent neural networks." In: *Proceedings of the 4th International Conference on Learning Representations*. OpenReview.net, San Juan, PR, USA, 1–10.

S. Hochreiter and J. Schmidhuber. Nov. 1997. "Long short-term memory." *Neural Computation*, 9, 8, (Nov. 1997), 1735–1780.

Z. Huang, Z. Sun, J. Liu, and Y. Ye. 2024. "Group-aware graph neural networks for sequential recommendation." *Information Sciences*, 670, 120623.

W.-C. Kang and J. McAuley. 2018. "Self-attentive sequential recommendation." In: *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, Singapore, 197–206.

J. Li, Y. Wang, and J. McAuley. 2020. "Time interval aware self-attention for sequential recommendation." In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM Press, New York, NY, USA, 322–330.

K. Liu, X. Zhao, J. Tang, W. Zeng, J. Liao, F. Tian, Q. Zheng, J. Huang, and A. Dai. 2021. "MOOPer: A large-scale dataset of practice-oriented online learning." In: *Knowledge Graph and Semantic Computing: 6th China Conference, CCKS 2021*. Springer-Verlag, Singapore, 281–287.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. "BLEU: A method for automatic evaluation of machine translation." In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, Philadelphia, PA, USA, 311–318.

B. Paudel and A. Bernstein. 2021. "Random walks with erasure: Diversifying personalized recommendations on social and information networks." In: *Proceedings of the Web Conference 2021*. ACM Press, New York, NY, USA, 2046–2057.

V. A. Traag, L. Waltman, and N. J. van Eck. 2019. "From Louvain to Leiden: guaranteeing well-connected communities." *Scientific Reports*, 9, 1, 5233.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. "Attention is all you need." In: *Advances in Neural Information Processing Systems 30*. Curran Associates Inc., Red Hook, NY, USA, 5998–6008.

H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. 2019. "Knowledge graph convolutional networks for recommender systems." In: *The World Wide Web Conference*. ACM Press, New York, NY, USA, 3307–3313.

P. Wang, Y. Fan, L. Xia, W.-X. Zhao, S. Niu, and J. Huang. 2020. "KERL: A knowledge-guided reinforcement learning model for sequential recommendation." In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, USA, 209–218.

X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. 2019. "KGAT: Knowledge graph attention network for recommendation." In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM Press, New York, NY, USA, 950–958.

W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang. 2024. "LLMRec: Large language models with graph augmentation for recommendation." In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. ACM Press, New York, NY, USA, 806–815.

S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie. 2024. "C-Pack: Packed resources for general Chinese embeddings." In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, USA, 641–649.

J. Yu et al.. 2020. "MOOCCube: A large-scale data repository for NLP applications in MOOCs." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, Stroudsburg, PA, USA, 3135–3142.

H. Zhang, X. Shen, B. Yi, W. Wang, and Y. Feng. 2023. "KGAN: Knowledge grouping aggregation network for course recommendation in MOOCs." *Expert Systems with Applications*, 211, 118344.

J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, and J. Sun. 2019. "Hierarchical reinforcement learning for course recommendation in MOOCs." In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA, USA, 435–442.

Z. Zheng, C. Wang, Z. Qiu, H. Zhu, and H. Xiong. 2024. "Harnessing large language models for text-rich sequential recommendation." In: *Proceedings of the ACM Web Conference 2024*. ACM Press, New York, NY, USA, 3207–3216.

T. Zhu, L. Sun, and G. Chen. 2023. "Graph-based embedding smoothing for sequential recommendation." *IEEE Transactions on Knowledge and Data Engineering*, 35, 1, 496–508.

X. Zhu, P. Zhao, J. Xu, J. Fang, L. Zhao, X. Xian, Z. Cui, and V. S. Sheng. 2020. "Knowledge graph attention network enhanced sequential recommendation." In: *Web and Big Data: 4th International Joint Conference, APWeb-WAIM 2020*. Springer-Verlag, Cham, Switzerland, 181–195.

Received 10 December 2025; accepted 24 April 2026