The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

GraphRAG-Induced Dual Knowledge Structure Graphs for Personalized Learning Path Recommendation

Xinghe Cheng[1], Zihan Zhang[1], Jiapu Wang[2], Liangda Fang[1]*,
Chaobo He[3], Quanlong Guan[1]†, Shirui Pan[4], Weiqi Luo[1]
[1]Jinan University
[2]Nanjing University of Science and Technology
[3]South China Normal University
[4]Griffith University
{chengxh, zhangzihan}@stu2023.jnu.edu.cn, jiapuwang9@gmail.com, hechaobo@m.scnu.edu.cn
{fangld, gql, lwq}@jnu.edu.cn, s.pan@griffith.edu.au

Abstract

Learning path recommendation seeks to provide students with a structured sequence of learning items (*e.g.*, knowledge concepts or exercises) to optimize their learning efficiency. Despite significant efforts in this area, most existing methods primarily rely on prerequisite relations, which present two major limitations: (1) Prerequisite relations between knowledge concepts are difficult to obtain due to the cost of expert annotation, hindering the application of current learning path recommendation methods. (2) Relying on a single sequentially dependent knowledge structure based on prerequisite relations implies that a confusing knowledge concept can disrupt subsequent learning processes, which is referred to as blocked learning. To address these two challenges, we propose a novel approach, GraphRAG-Induced Dual Knowledge Structure Graphs for Personalized Learning Path Recommendation (KnowLP), which enhances learning path recommendations by incorporating both prerequisite and similarity relations between knowledge concepts. Specifically, we introduce a knowledge structure graph generation module EDU-GraphRAG that constructs knowledge structure graphs for different educational datasets, significantly improving the applicability of learning path recommendation methods. We then propose a Discrimination Learning-driven Reinforcement Learning (DLRL) module that utilizes similarity relations as fallback relations when prerequisite relations become ineffective, thereby alleviating the blocked learning. Finally, we conduct extensive experiments on three benchmark datasets, demonstrating that our method not only achieves state-of-the-art performance but also generates more effective and longer learning paths.

Code — https://github.com/chanllon/KnowLP

Introduction

With the rapid advancement of online education platforms, there is a growing demand for personalized learning experiences tailored to individual students. In this context, Learning Path Recommendation (LPR) has emerged as a critical task, aiming to construct coherent and customized sequences of learning resources that align with specific educational goals (Wang et al. 2023a; Zhang, Lu, and Zhang 2021; Chen 2009; Shi et al. 2020). By arranging content in a pedagogically meaningful order, LPR facilitates systematic and progressive knowledge acquisition, thereby improving learning efficiency and effectiveness (Nabizadeh, Mário Jorge, and Paulo Leal 2017).

Prior research in education has demonstrated that the structure of Knowledge Concepts (KCs) plays a pivotal role in effective learning path recommendations (Liu et al. 2019). Existing LPR approaches that incorporate KC structure can be categorized into two paradigms: (1) correlation-based methods (Chen et al. 2023), which infer personalized learning paths by exploiting statistical associations among KCs; and (2) prerequisite-based methods (Li et al. 2024; Zhang et al. 2024), which utilize prerequisite relations to guide the sequential organization of learning content. The prerequisite-based approaches have attracted increasing attention due to their strong pedagogical theoretical foundation and consistency with human cognitive progression, leading to more interpretable and cognitively coherent learning paths.

Prerequisite relations among KCs serve as a structural foundation for learning path recommendation, enabling stepwise progression toward learning objectives. However, the construction of such relations often relies on costly expert annotations, thereby limiting the applicability of prerequisite-based methods (Zhang et al. 2024). Even when prerequisite graphs are available, they frequently suffer from data sparsity and noise due to missing concepts or inaccurate dependencies. Recent advances in Large Language Models (LLMs) offer a compelling alternative by inferring prerequisite structures from natural language. Yet, their effectiveness remains constrained by the limited availability of high-quality textual descriptions in practice and the inherent risk of generating incorrect dependencies.

One of the crucial reasons that students fail to master a KC c is that they are confused by some KCs similar to c. As a result, when relying on a single sequentially dependent knowledge structure based on prerequisite relations, a confusing KC can disrupt subsequent learning processes

---

*Corresponding author

†Corresponding author

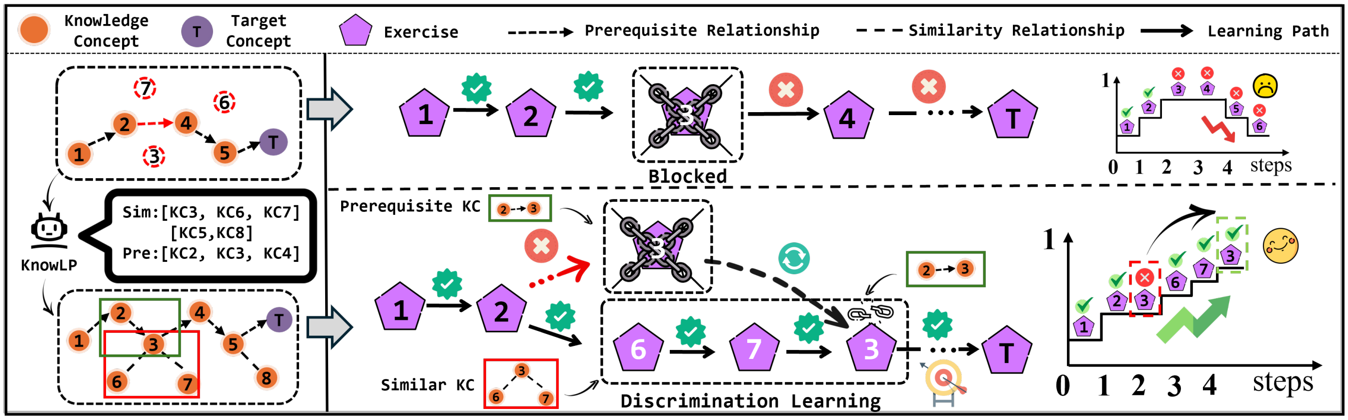


Figure 1: The comparison of traditional methods (top) and our proposed KnowLP (bottom) is depicted. The top denotes traditional prerequisite relation-based methods where confusion between similar KCs such as KC_3, KC_6, and KC_7 can hinder students' mastery of KC_3 and impede their understanding of subsequent KCs. The bottom illustrates how KnowLP enables students to more effectively differentiate between these similar KCs, thereby improving learning outcomes.

(Pan, Selvarajan, and Murphy 2024). This phenomenon is referred to as blocked learning. As shown in the top part of Figure 1, merely using prerequisite relations struggles to handle blocked learning effectively. Gagné's learning hierarchy theory (Gagné 1968) introduces the concept of discrimination learning, which supports students in mastering confuse KCs by building on their knowledge of similar KCs (Chinda 2022; Rohrer 2012). As illustrated in the bottom part of Figure 1, we demonstrate that the student will be able to progress smoothly after learning some KCs similar to the confusing one.

Therefore, LPR faces two significant challenges: (1) inferring accurate similarity and prerequisite relations among KCs within different datasets, and (2) determining when to introduce similar KCs during learning path generation. To address the aforementioned two challenges, we make several contributions in this paper.

1. We propose a GraphRAG-Induced Dual Knowledge Structure Graphs for Personalized Learning Path Recommendation (KnowLP) framework, a novel method that jointly incorporates prerequisite and similarity relations between KCs to construct more effective learning paths.
2. To adaptively generate prerequisite and similarity relation graphs for KCs on different datasets, we introduce a knowledge structure graph generation module EDU-GraphRAG.
3. Based on the generated graphs, we design a Discrimination Learning-driven Reinforcement Learning (DLRL) module that utilizes similarity relations as fallback relations when prerequisite relations become ineffective, thereby alleviating the blocked learning.
4. We propose three agents: prerequisite, similarity and difficulty to simulate the discrimination learning process. The prerequisite agent decides whether the learning path should conform to prerequisite dependencies. The similarity agent detects conceptually confused KCs in blocked learning scenarios. The difficulty agent selects exercises of appropriate difficulty and integrates them into the learning path based on these KCs.
5. We conduct extensive experiments on three educational datasets, demonstrating that our method not only achieves state-of-the-art performance but also generates more effective longer learning paths.

Related Work

Learning Path Recommendation Existing methods can be categorized into two types according to the approach of KC structures considered: correlation-based methods and prerequisite relation-based methods.

For correlation-based approaches, many works have been proposed using different techniques, such as decision tree classifiers (Lin et al. 2013), matrix factorization (Nabizadeh et al. 2020), Bayes' theorem (Xu et al. 2012), knowledge graph (Guan et al. 2025; Wang et al. 2023b), to model the temporal correlation features in the learning process. Notably, the **S**et-to-Sequence **R**anking-Based **C**oncept-Aware Learning Path Recommendation (SRC) framework proposed by Chen et al. (2023) has gained prominence for modeling the complex relations and interactions among KCs, enabling the generation of complete learning paths. Due to their ability to better guide the sequential organization of learning content (Ma et al. 2023), prerequisite relation-based methods are rapidly gaining attention. These approaches frequently treat learning path recommendation as a sequential decision-making problem, which results in the frequent application of advanced reinforcement learning techniques (Intayoad, Kamyod, and Temdee 2020; Liu et al. 2019; Zhang et al. 2024). Despite significant efforts in this area, none of these related works have attempted to focus on incomplete prerequisite relations and similarity relations, leading to inappropriate learning path recommendations.

Retrieval-Augmented Generation Retrieval-Augmented Generation (RAG) is a well-established method for answering user queries across large datasets(Ram et al. 2023;

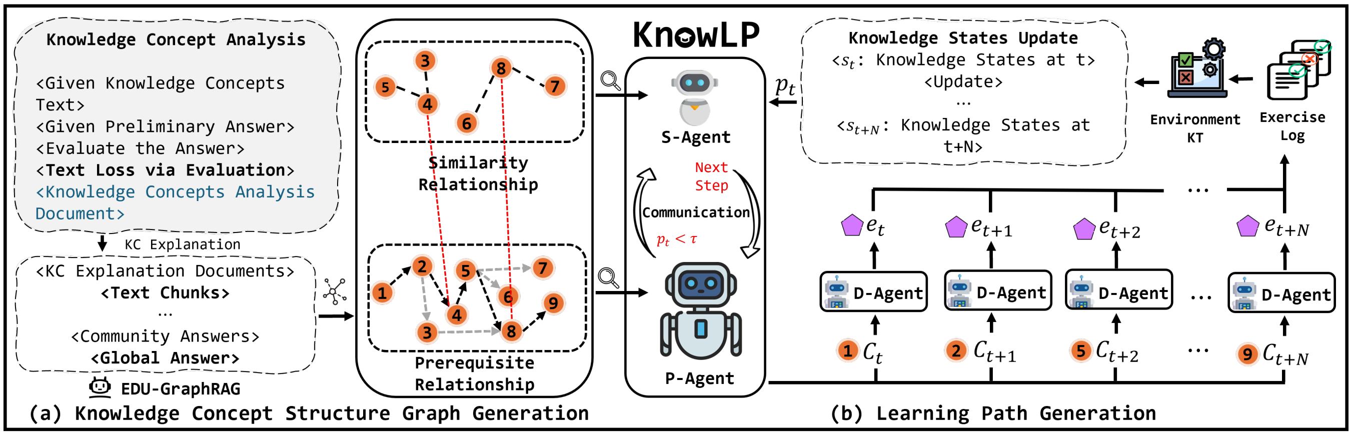


Figure 2: KnowLP Framework Overview. The knowledge structure graph generation module on the left begins by inputting KC names into TextGrad, which then generates textual descriptions. These descriptions are subsequently used by the EDU-GraphRAG to construct knowledge structure graphs. On the right, the learning path generation module initiates with the prerequisite agent (P-Agent) and similarity agent (S-Agent) filtering KCs. Following this, the difficulty agent (D-Agent) selects exercises of appropriate difficulty. The module then utilizes an environment KT to dynamically update the knowledge state in each session. Finally, the module generates the learning path step by step.

Laskar, Hoque, and Huang 2020; Goodwin, Savery, and Demner-Fushman 2020). When applying large language models (LLMs), RAG first retrieves relevant information from external data sources and then incorporates this information into the context window of LLM along with the original query (Ram et al. 2023; Lewis et al. 2020). In contrast to other RAG approaches, Edge et al. (2024) proposed a GraphRAG method that leverages global summarization of an LLM-derived knowledge graph (Wang et al. 2024b), highlighting a previously unexplored aspect of graph structures in this context (Zhang et al. 2025; Wang et al. 2024a; Meng et al. 2024). However, GraphRAG methods typically require sufficiently accurate documentation for effective implementation.

Preliminaries

In this section, we introduce the problem of learning path recommendations and the technique of graph retrieval-augmented generation.

Learning Path Recommendation The primary objective of this paper is to address learning path recommendation problem. Suppose we have a set of exercises $E = \{e_1, e_2, \cdots, e_M\}$, a set of KCs $C = \{c_1, c_2, \cdots, c_N\}$ and a set of learning goals $G = \{g_1, g_2, \cdots, g_K\}$ where each $g_i \in G$ denotes the learning goal of student s_i. Each historical record $h = (e, s)$ consists of an exercise $e \in E$ and its corresponding score s. A learning path $p = \{\tilde{e}_1, \tilde{e}_2, \cdots, \tilde{e}_k\}$ where each $\tilde{e}_i \in E$ is a recommended exercise. To evaluate the effectiveness of a learning path for student s_i, we calculate the improvement of scores for s_i between the beginning and the end of the session. The effectiveness of the learning path recommendation, denoted as E_i, is computed as follows:

$$E_i = \frac{S_i^k - S_i^0}{S_i^{\text{sup}} - S_i^0}, \tag{1}$$

where S_i^{sup} represents the maximum achievable score for student s_i and S_i^0 (resp. S_i^k) denote the mastery level of the target concepts of student s_i before (resp. after) the learning path p.

Definition 1 Given a sequence of records $H = \{h_1, h_2, \cdots, h_j\}$ and a learning goal $g \in G$, the goal of learning path recommendation is to generate a learning path that maximize the E_i.

Graph Retrieval-Augmented Generation Given a natural language query Q and a source document D, GraphRAG (Edge et al. 2024) aims to transform unstructured textual content into a structured graph representation $\phi(D)$, which facilitates accurate and explainable answer generation. To achieve this, we first to derive an entity knowledge graph $\mathcal{G}(\cdot, \cdot)$ from the source documents. At query time, the model generates the final output A by conditioning an LLM on the query Q and the structured graph index $\phi(D)$:

$$A = \text{LLM}(Q, \phi(D)) = \text{LLM}(Q, \bigcup_{i=1}^{n} \mathcal{G}(\mathcal{E}, \mathcal{R})), \tag{2}$$

where $\mathcal{E}$ denotes the set of extracted entities, $\mathcal{R}$ denotes the set of corresponding relations among them, and n denotes the number of chunks.

Methodology

In this section, we introduce our approach, KnowLP, as illustrated in Figure 2, which consists of two main components: knowledge structure graph generator and

Prompt for KC Explanation Generation

LLM: (“You are an expert in the filed of education and are responsible for the detailed interpretation of knowledge concepts. The following are the knowledge concepts. Each knowledge concept is separated by ‘,’:knowledge concepts.
Question: Use the knowledge in education to analyze the meaning of the serial number is begin to end. Knowledge concepts are: knowledge concepts[begin:end+1], as well as detailed analysis the relation between each of these knowledge concepts and other knowledge concepts.
Output Format: [knowledge concept]:[The analysis of knowledge concept.]
Evaluation Instruction: Make the generated analysis smarter, more logical and accurate, more specific and discriminative rather than vague, and avoid ambiguity. Ensure that the analysis of knowledge concepts is correct.”)

learning path generator. Knowledge structure graph generator synthesizes high-quality knowledge structure graphs by adaptively discovering both prerequisite and similarity relations across different datasets. Learning path generator constructs effective and coherent learning paths by simulating the discrimination learning process.

Knowledge Structure Graph Generator

In this subsection, we present an overview of the knowledge structure graph generator. Knowledge structure graph generator consists of two operations: KC explanation generator and EDU-GraphRAG. By leveraging the TextGrad framework (Yuksekgonul et al. 2025), KC explanation generator synthesizes explanations for each knowledge concept (KC) in an iterative way. Based on these KC explanations, we design an EDU-GraphRAG framework that automatically constructs knowledge structure graphs, which serve as a crucial foundation for learning path recommendation.

Knowledge Concept Explanation Generator We generalize Textgrad to improve the reliability of KC explanations by an iterative generation–evaluation–refinement procedure. With prompt engineering, we guide the LLM to act as a domain expert in education. We first use a main LLM, denoted by LLM_1, with predefined prompt P_{gen} to generate an initial explanation $\mathcal{T}^{(0)}$. For each iteration t, we use an auxiliary LLM_2 with evaluation prompt P_{eval} to evaluate explanation $\mathcal{T}^{(t)}$, which produces feedback $\nabla^{(t)}_{\mathrm{LLM}_2}$ via an additional feedback prompt P_{feedback}. Lastly, the explanation is refined with a rewriting prompt P_{rewrite} as:

$$\nabla^{(t)}_{\mathrm{LLM}_2} = \mathrm{LLM}_2\left(P_{\text{feedback}}, \mathrm{LLM}_1\left(P_{\text{eval}}, c, \mathcal{T}^{(t)}\right)\right), \tag{3}$$

$$\mathcal{T}^{(t+1)} = \mathrm{LLM}_1\left(P_{\text{rewrite}}, \mathcal{T}^{(t)}, \nabla^{(t)}_{\mathrm{LLM}_2}\right). \tag{4}$$

This iterative process continues until it converges or a predefined maximum step T is reached. We take the finalized KC explanation as the source document D_c, to be consumed in the next stage of our framework. For instance, the objective of the KC explanation is shown in the box Prompt for KC Explanation Generation.

EDU-GraphRAG Given a collection of N concept-level explanation documents $\{D_{c_1}, D_{c_2}, \ldots, D_{c_N}\}$, where each D_{c_i} provides a contextualized explanation of a distinct knowledge concept c_i, we construct a unified document D_c by concatenating all these explanations in sequence. The document D_c is segmented into a sequence of text chunks using a parameterized segmentation function $S(D_c; \theta) = \{\sigma_1(D_c; \theta), \ldots, \sigma_n(D_c; \theta)\}$, where S denotes a parameterized segmentation function, θ is the chunk size, and $\sigma_i(D_c; \theta)$ represents the i-th chunk. For each chunk, an LLM extracts entities and relations, capturing both explicit semantics in the text. Then, we derive an entity knowledge graph $\mathcal{G}(\cdot, \cdot)$ from the source documents D_c. The global document-level graph is obtained by unifying these knowledge graphs:

$$\phi(D_c) = \bigcup_{i=1}^{n} \mathcal{G}\left(\mathcal{E}(\sigma_i(D_c; \theta)), \mathcal{R}(\sigma_i(D_c; \theta))\right). \tag{5}$$

At query time, given a query Q, EDU-GraphRAG conditions an LLM on the structured graph index $\phi(D_c)$ to generate the final output $A_{\mathcal{G}}$, which in our case corresponds to a constructed knowledge structure graph $\mathcal{G}$.

$$\mathcal{G}_A = \mathrm{LLM}(Q, \phi(D_c)), \tag{6}$$

where $\mathcal{G}_A = (\mathcal{C}, \mathcal{P}, \mathcal{S})$ with KC set $\mathcal{C}$, prerequisite relation $\mathcal{P}$ and similarity relation $\mathcal{S}$.

Learning Path Generator

In this subsection, we introduce the *Discrimination Learning-driven Reinforcement Learning (DLRL)* module, which dynamically generates personalized learning paths while alleviating blocked learning. To model the student’s evolving knowledge state, we employ *Difficulty Matching Knowledge Tracing (DIMKT)* (Shen et al. 2022) that takes into account the influence of exercise difficulty on students’ cognitive changes. Then, we propose three agents: prerequisite, similarity and difficulty to simulate the discrimination learning process. The prerequisite agent decides whether the learning path should conform to prerequisite dependencies. The similarity agent detects confused KCs in blocked learning scenarios. The difficulty agent selects exercises of appropriate difficulty and integrates them into the learning path based on these KCs.

Knowledge Tracing Knowledge tracing algorithms track the evolving knowledge states of students based on their past learning interactions. By leveraging this technique to devise subsequent recommendation strategies, we provide a learning path that matches the student’s abilities and progress. DIMKT is advantageous as it can predict the impact of exercise difficulty on students, thereby enhancing the accuracy of recommending exercises based on their difficulty level.

In this process, we employ the same Heterogeneous Graph Neural Network (HGNN) (Yang et al. 2023; Zhang, Yuan, and Pan 2024; Zheng et al. 2023) embedding technique as utilized in Difficulty-constrained Learning Path Recommendation (DLPR) (Zhang et al. 2024) to embed both exercises and difficulty levels as inputs to DIMKT. This method effectively integrates exercise features with difficulty vectors, ensuring that the predicted knowledge states comprehensively account for the interrelations between exercises and their corresponding difficulty levels.

Prerequisite Agent To ensure that the generated learning path follows a pedagogically appropriate order for the student, we develop the prerequisite agent to select KCs sequentially based on their prerequisite relations.

(1) State encoder: At each step, the predicted knowledge state is based on the student's learning objectives and historical learning records. Thus, after step $t-1$, the state of the prerequisite agent includes the student's knowledge state $\mathbf{h}_{t-1}$ and learning objectives $\mathbf{G}$. Specifically, we use a one-hot encoding scheme to represent the learning objective $\mathbf{G} = \{0,1\}^N$, where N is the total number of KCs. The index corresponding to the learning objective is set to 1, while all other indices are set to 0. At step t, the state of the prerequisite agent $\mathbf{s}_t$ is represented as follows:

$$\mathbf{s}_t = \mathbf{h}_{t-1} \oplus \mathbf{G}. \tag{7}$$

(2) Reward: To maximize the improvement in students' mastery of the learning goals, we calculate the corresponding reward value after selecting KC at each step.

$$Re_t = \begin{cases} E_i^t, & \text{if } t \text{ is the last learning stage;} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

(3) Policy: We adopt Proximal Policy Optimization (PPO) (Schulman et al. 2017) as the model for the prerequisite agent since it is one of the most effective reinforcement learning models. PPO is effective at identifying the optimal sequence of KCs to achieve the learning goal. PPO consists of two primary components: a policy network (Actor) $\pi(\mathbf{G}\,|\mathbf{s}_t;\theta)$ and a value network (Critic) $V(\mathbf{s}_t;\phi)$, where θ and ϕ represent the respective network parameters.

The goal of the policy network is to generate a probability distribution over the KCs in the dataset. The state of the prerequisite agent is fed into the policy network, which outputs a corresponding probability distribution over the KCs.

$$\mathbf{G} \sim \pi(\mathbf{G}\,|\mathbf{s}_t;\theta) = \mathrm{Softmax}(\mathrm{Linear}(\mathbf{s}_t)), \tag{9}$$

where Linear is the fully connected layer. The value network takes the state of the agent as input and evaluates the reward associated with that state.

$$V(\mathbf{s}_t;\phi) = \mathrm{Linear}(\mathbf{s}_t). \tag{10}$$

During each training step, we optimize PPO using a loss function similar to Graph Enhanced Hierarchical Reinforcement Learning (GEHRL) (Chen et al. 2023). The value network is trained using the mean squared error (MSE) loss function with gradient descent.

$$L(\phi) = \mathbb{E}\left(\left\|\sum_{i=0}^{T-t} \gamma^i Re_{t+i} - V(\mathbf{s}_t;\phi)\right\|\right)^2, \tag{11}$$

where Re_t represents the reward at step t and γ denotes the discount factor. We optimize the policy network using PPO-clip (Chen et al. 2023).

$$\hat{A}_t = -V(\mathbf{s}_t;\phi) + Re_t + \cdots + \gamma^{T-t} V(\mathbf{s}_T;\phi), \tag{12}$$

$$L(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{13}$$

where $r_t(\theta)$ represents the probability ratio $\frac{\pi(G|s_t;\theta)}{\pi(G|s_t;\theta_{old})}$, $\epsilon = 0.2$, and $\mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$ modifies the surrogate objective by clipping the probability ratio, which removes the incentive for moving r_t outside of the interval $[1-\epsilon, 1+\epsilon]$.

Similarity Agent When relying on a single sequentially dependent knowledge structure based on prerequisite relations, a student's inability to master a difficult KC can disrupt subsequent learning processes (Pan, Selvarajan, and Murphy 2024). To address this issue, we introduce a similarity agent that uses the generated similarity relation graph to simulate the discrimination learning process. The similarity agent identifies a set of similar KCs that are related to the current KC c_t. From this set, the similarity agent selects a sub-path that can effectively enhance the student's mastery of the current KC. This sub-path is then appended to the learning path generated by the prerequisite agent. The key role of the similarity agent is activated when the prerequisite agent fails to identify an appropriate next KC. In such cases, the similarity agent leverages the generated similarity relation graph to search confusable KCs and help students discriminate them in the learning path, thereby alleviating blocked learning. The similarity agent operates identically to the prerequisite agent in terms of architecture except that it takes similarity relations as input.

Difficulty Agent To recommend exercises of appropriate difficulty, which lead to a loss of interest during the learning process (Minn, Zhu, and Desmarais 2018; Zhang et al. 2022). We aim to find the exercise e_{m^*} whose difficulty most closely matches the student's mastery of KC c_i.

$$m^* = \arg\min_{m \in M} |\mathrm{diff}(e_m) - h^{c_i}|, \tag{14}$$

where m^* represents the exercise ID that best matches the difficulty for the student, M is the set of exercises that is related to KC c_i, $|\cdot|$ represents the absolute value, $\mathrm{diff}(e_m)$ represents the difficulty of the exercise e_m, and h^{c_i} indicates the student's mastery of KC c_i.

Learning Path Construction Mechanism In this section, we introduce the strategy for selecting initial nodes as well as the communication mechanism among agents.

(1) Node Initialization: Due to the vast number of KCs, this selection process results in a large search space. To reduce this complexity, it is essential to identify a shortest learning path between an initial node and the learning objective before the prerequisite agent performs its selection. Existing advanced methods, such as adaptive action space learning algorithms, employ the A* algorithm (Duchoň et al. 2014) to generate the shortest paths while dynamically determining candidate action spaces. Nonetheless, these

methods often fail to identify an effective initial node, which may result in the selection of unsuitable KCs, leading to an ineffective learning path.

To address this limitation, we introduce an initial node identification algorithm to ensure the selection of the most appropriate initial node. Specifically, our approach begins from the target KC node and iteratively traces back along the prerequisite relation. During this backward traversal, we assess whether the prerequisite KCs of the current node have already been mastered by the student. If these prerequisite concepts are mastered, the backward traversal terminates. Otherwise, we continue tracing back to the node with the longest chain of prerequisite KCs. By incorporating this method, the most suitable initial node can be determined before dynamically identifying the candidate action space, effectively optimizing the overall learning path generation.

(2) Agent Switching: At step t, the prerequisite agent evaluates the improvement in the student's mastery of prerequisite KC c_{t-1} when selecting the next KC c_t to recommend. The evaluation is based on the following:

$$p_t = h_{t-1}^{c_t} - h_{t-2}^{c_t}, \tag{15}$$

where $h_{t-1}^{c_t}$ represents the student's knowledge mastery level associated with the KC c_t learned at step $t-1$, and $h_{t-2}^{c_t}$ represents the knowledge mastery level at step $t-2$. If the improvement p_t is below a predefined threshold τ, the similarity agent is activated. Once the similarity agent completes its selection process, the prerequisite agent resumes selecting subsequent KCs based on their prerequisite relations. After the prerequisite agent and similarity agent select the KCs, we incorporate a difficulty agent to match the selected concepts with exercises that align with the student's knowledge mastery level.

Experiments

In this section, we describe the experimental setup and conduct a comprehensive analysis of the results.

Experimental Setup

Datasets We conduct our experiments using three publicly available educational datasets to validate the effectiveness of our model: Junyi [1], MOOCCubeX [2] (MCX), and ASSISTments2009 [3] (ASS09) . The Junyi dataset provides a comprehensive knowledge structure graph, illustrating the prerequisite relations among knowledge concepts. In contrast, the MOOCCubex dataset contains an incomplete knowledge structure graph, while the ASSIST09 dataset lacks a knowledge structure graph altogether. Furthermore, none of the datasets include information on the similarity relations between KCs. However, all three datasets contain knowledge concept names, allowing us to construct knowledge structure graphs for both prerequisite and similarity relations using our method. The statistical details of the datasets are shown in Table 1.

Dataset	Junyi	MCX	ASS09
#KCs	835	443	167
#students	525,061	629	4,217
#Records	21,460,249	17,447	346,860
Positive label rate	54.38%	70.18%	63.81%

Table 1: Dataset Statistics.

Baselines We compare KnowLP with the following baselines:

- KNN (Cover and Hart 1967): The K-Nearest Neighbors algorithm identifies the K closest neighboring values to the target. We use the cosine similarity between learning paths to measure the similarity between two students and recommend the most similar learning paths.
- GRU4Rec (Hidasi 2015): GRU4Rec is a session-based recommendation model based on Gated Recurrent Units (GRU). It trains an RNN model to capture patterns and information within sequences, predicting the probability of the next exercise appearing.
- Actor-Critic (Konda and Tsitsiklis 1999): This method encodes input data using a GRU and then employs a vanilla actor-critic framework for learning path recommendation.
- RL-Tutor (Kubotani, Fukuhara, and Morishima 2021): RL-Tutor is an RL-based adaptive tutoring system that simulates student behavior to recommend learning paths.
- CSEAL (Liu et al. 2019): CSEAL uses Deep Knowledge Tacing (DKT) (Piech et al. 2015) to estimate knowledge states and designs a cognitive navigation algorithm to guide the vanilla actor-critic framework in recommending learning paths.
- SRC (Chen et al. 2023): SRC uses a concept-aware encoder to capture the relations between knowledge concepts, and then a decoder generates learning paths.
- GEHRL (Li et al. 2023): GEHRL employs a hierarchical reinforcement learning approach, where the lower-level agent filters relevant knowledge concepts for the target, while the higher-level agent is responsible for planning the sequence.
- DLPR (Zhang et al. 2024): DLPR uses a hierarchical graph neural network to aggregate information about learning tasks and difficulty and then applies a hierarchical reinforcement learning framework to plan knowledge concepts and recommend exercises.

Implementation Detail In the process of generating the KC structure graph, we adjust the chunk size based on the number of KCs in the dataset. The chunk sizes used for the Junyi, ASS09, and MCX datasets were 800, 600, and 400, respectively. In the process of recommending learning paths, the first 60% of exercise logs in the dataset were used to train the environment KT, while the remaining exercise records were used to train and test our model. All experiments are conducted on an NVIDIA GeForce RTX 4090.

---

[1] https://pslcdatashop.web.cmu.edu

[2] https://github.com/THU-KEG/MOOCCubeX

[3] https://sites.google.com/site/assistmentsdata

Dataset	Steps	KNN	GRU4Rec	RL-Tutor	Actor-Critic	CSEAL	SRC	GEHRL	DLPR	KnowLP
Junyi	5	0.1296	0.1504	0.1601	0.1891	0.1964	0.1874	0.1762	0.2086	**0.2406***
	10	0.1485	0.2129	0.1923	0.2051	0.1779	0.2075	0.2222	0.2293	**0.2431***
	15	0.1769	0.1714	0.1272	0.1428	0.1981	0.1368	0.2105	0.2181	**0.2295***
	20	0.1415	0.1909	0.1052	0.2129	0.1880	0.1739	0.2358	0.1880	**0.2758***
MCX	5	0.2131	0.2500	0.2500	0.2501	0.2686	0.2131	0.1164	0.2388	**0.3194***
	10	0.1304	0.1846	0.2001	0.2089	0.1718	0.2203	0.1342	0.2173	**0.3508***
	15	0.1791	0.2343	0.1875	0.1818	0.2328	0.2695	0.1689	0.2933	**0.3205***
	20	0.2388	0.2272	0.1600	0.2352	0.2676	0.2631	0.1063	0.2781	**0.3589***
ASS09	5	0.0504	0.0771	0.0625	0.0498	0.0971	0.0759	0.0966	0.1245	**0.1268***
	10	0.0916	0.0849	0.0750	0.0696	0.1024	0.0760	0.1018	0.0926	**0.1174***
	15	0.0734	0.0482	0.0843	0.0836	0.0931	0.1273	0.1183	0.0996	**0.1281***
	20	0.0666	0.0675	0.1111	0.0616	0.0769	0.1417	0.1195	0.1434	**0.1455***

Table 2: Performance comparison for learning path recommendation methods. Existing state-of-the-art results are underlined and the best results are bold. Our KnowLP is compared with the SOTA DLPR and $*$ indicates a p-value< 0.05 in the t-test.

Experimental Results

In this section, we validate the effectiveness of KnowLP by answering the following key research questions (RQs):

- RQ1: How does KnowLP perform compared to various state-of-the-art LPR models?
- RQ2: Is the similarity agent effective in improving the performance?
- RQ3: Is EDU-GraphRAG effective in knowledge structure graph genration?
- RQ4: How effectively does TextGrad enhance KnowLP, and does KnowLP generate reasonable reasons for recommended learning paths?
- RQ5: How effective is KnowLP in simulation experiments?
- RQ6: What is the runtime of the proposed KnowLP?
- RQ7: How does different hyperparameter τ affect the performance?

Overall Performance (RQ1) Table 2 presents the overall performance of all methods across the three datasets, revealing several key insights. In comparison with all baseline models, our method outperforms the others on all three datasets. This demonstrates that leveraging LLM-generated knowledge structure graphs, combined with the integration of similarity relations, contributes significantly to improving recommendation performance. Specifically, EDU-GraphRAG effectively mitigates the hallucination problem associated with knowledge structure graphs generated by LLMs, which then serve as reliable guidance for learning path recommendation. Moreover, incorporating similarity relations facilitates smoother learning progression, thereby enhancing the overall performance.

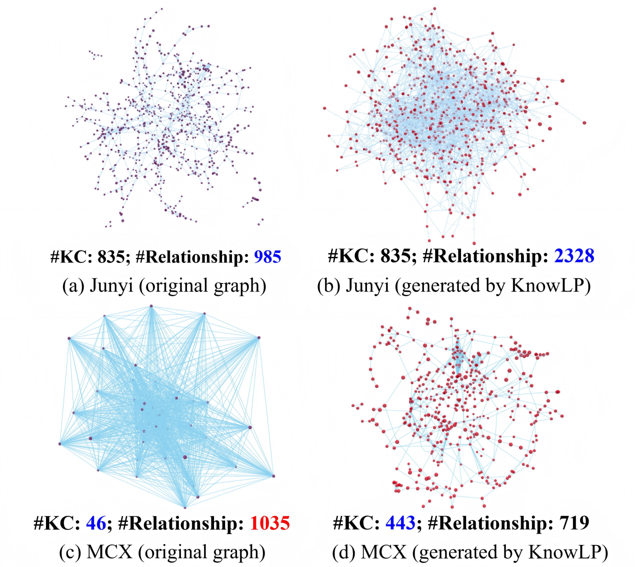
Furthermore, most baseline methods exhibit a decline in performance or fail to achieve their best results when the recommendation step reaches 20. In contrast, KnowLP consistently achieves optimal performance at step 20 across all three datasets. This further demonstrates that the similarity-aware design effectively smooths the learning process.

Dataset/Steps	**5**	**10**	**15**	**20**
Junyi (w/o S)	0.1932	0.2051	0.1851	0.2232
Junyi (w S)	**0.2406**	**0.2431**	**0.2295**	**0.2758**
MCX (w/o S)	0.2148	0.1928	0.2258	0.1517
MCX (w S)	**0.3194**	**0.3508**	**0.3205**	**0.3589**
ASS09 (w/o S)	0.0737	0.1007	0.1190	0.1264
ASS09 (w S)	**0.1268**	**0.1174**	**0.1281**	**0.1455**

Table 3: Ablation Experiment: "w/o S" represents the model without similarity agent.

Ablation Study (RQ2) To validate the effectiveness of the innovative components of our method, we conducted an ablation study. Specifically, we remove the similarity agent module from our method to assess its contribution to the overall performance. As shown in Table 3, when the similarity agent is removed, there is a noticeable performance drop across all three datasets. This highlights the necessity and value of considering the similarity relations in learning path recommendations. Moreover, with the similarity agent included, our model achieves a substantial performance improvement at step 20. This further confirms that similarity relations play a critical role in unlocking the performance potential of longer learning paths.

Knowledge Structure Graph Generation (RQ3) To demonstrate the effectiveness of the prerequisite relation graph generated by KnowLP, we visualize both the original and generated knowledge structure graphs for the Junyi and MCX datasets. The visualization results of the prerequisite relation graphs are shown in Figure 3. It is important to note that since ASS09 does not explicitly provide the prerequisite relation graph, it is not included in the analysis. We observe that in the Junyi, the number of relations in the generated prerequisite relation graph is higher than in the original graph, indicating that KnowLP captures more complex prerequisite relations between KCs. In the MCX, the orig-



#KC: 835; #Relationship: 985
(a) Junyi (original graph)

#KC: 835; #Relationship: 2328
(b) Junyi (generated by KnowLP)

#KC: 46; #Relationship: 1035
(c) MCX (original graph)

#KC: 443; #Relationship: 719
(d) MCX (generated by KnowLP)

Figure 3: Comparison between the original graph and the graph generated by KnowLP.

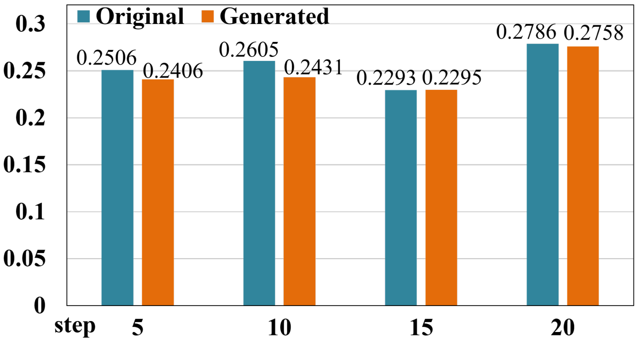
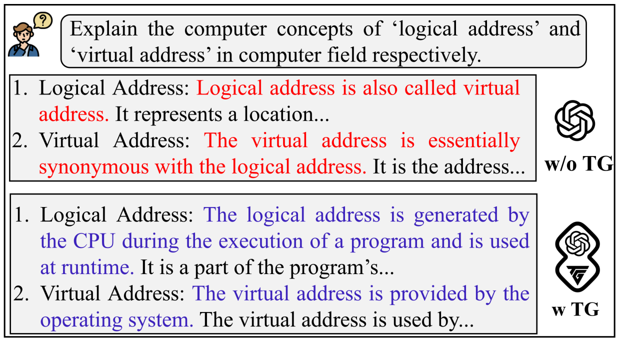


Figure 4: Performance comparison of the original graph and the KnowLP-generated graph.

inal knowledge structure graph, as shown in Figure 3 (c), covers only a small portion of the 443 KCs in the dataset. Specifically, it includes 1,035 relations among just 46 KCs, forming a fully connected graph in which every pair of KCs is linked. However, the relations offer limited utility due to the absence of meaningful prerequisite relations. In contrast, the graph generated by KnowLP incorporates all KCs and maintains a larger number of relations.

Furthermore, we compare the performance of models using the original prerequisite relation graph and the graph generated by our method on the Junyi dataset. Similarity relation graphs are uniformly generated by KnowLP. From Figure 4, we find that the performance of the model using the generated prerequisite relation graph is close to that using the original prerequisite relation graph. This demonstrates that, in addition to automatically generating the prerequisite relation graph, our method can produce prerequisite relation graphs that perform at a similar level to the original graph, which was manually constructed by experts based on domain knowledge and a deep understanding of the relations between KCs.

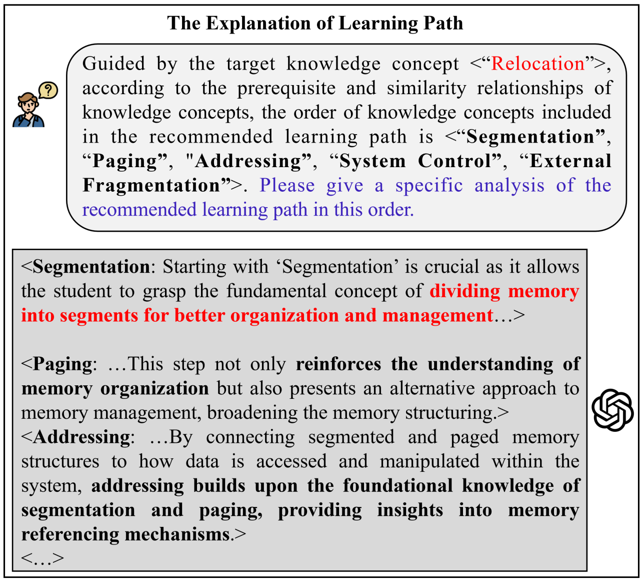


Explain the computer concepts of ‘logical address’ and ‘virtual address’ in computer field respectively.

1. Logical Address: Logical address is also called virtual address. It represents a location...
2. Virtual Address: The virtual address is essentially synonymous with the logical address. It is the address...

1. Logical Address: The logical address is generated by the CPU during the execution of a program and is used at runtime. It is a part of the program’s...
2. Virtual Address: The virtual address is provided by the operating system. The virtual address is used by...

Figure 5: Comparison of the results of with/without TextGrad (TG).



The Explanation of Learning Path

Guided by the target knowledge concept <“Relocation”>, according to the prerequisite and similarity relationships of knowledge concepts, the order of knowledge concepts included in the recommended learning path is <“**Segmentation**”, “**Paging**”, "**Addressing**", “**System Control**”, “**External Fragmentation**”>. Please give a specific analysis of the recommended learning path in this order.

<**Segmentation**: Starting with ‘Segmentation’ is crucial as it allows the student to grasp the fundamental concept of **dividing memory into segments for better organization and management**…>

<**Paging**: …This step not only **reinforces the understanding of memory organization** but also presents an alternative approach to memory management, broadening the memory structuring.>
<**Addressing**: …By connecting segmented and paged memory structures to how data is accessed and manipulated within the system, **addressing builds upon the foundational knowledge of segmentation and paging, providing insights into memory referencing mechanisms**.>
<…>

Figure 6: The explanation of learning path.

Case Study (RQ4) Figure 5 shows the explanation outputs of the KCs by the LLM (gpt-4o) before and after using TextGrad. When explaining the KCs “logical address” and “virtual address” using the LLM alone, confusion arises, as these two KCs are easily mixed up. The LLM would incorrectly explain that the “logical address” is the same as the “virtual address” . After optimizing the explanation process using the TextGrad method, it is clear that the LLM can effectively distinguish the two KCs.

In the process of generating the knowledge structure graph, EDU-GraphRAG also generates a community summary containing key entities, relations, and claims, which provides useful contextual information for subsequent queries. Taking advantage of this feature of EDU-GraphRAG, we are able to explain the reason for recommended learning paths, which improves the interpretability of the recommendations and enhances

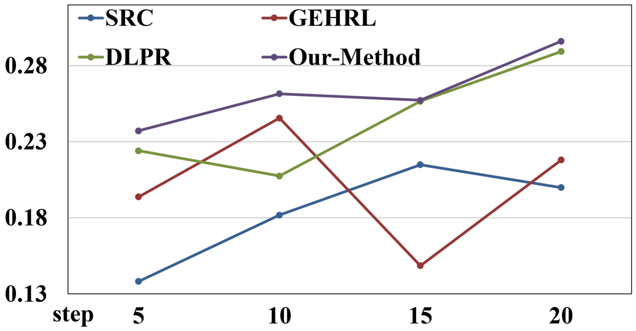


Figure 7: Results of simulation experiment.

students' trust in the recommended content. Figure 6 shows the analysis and explanation of the recommended paths by EDU-GraphRAG. For each recommended exercise, EDU-GraphRAG is able to explain the reason behind the recommendation of the corresponding KC.

Simulation Experiment (RQ5) Since real-world data only contains static information, which cannot be directly used to analyze the exercise sequences of students not included in the dataset, the effectiveness of our method in real online education scenarios cannot be directly validated. Therefore, we use a Knowledge Evolution-based Simulator (KES), which leverages the DKT model (Piech et al. 2015) to simulate students' exercise behavior on randomly generated exercise sequences. Subsequently, our method performs learning path recommendations based on the student behavior generated by the simulator to validate its effectiveness in real online education scenarios.

We construct the KES-Junyi simulator (Liu et al. 2019) based on the Junyi dataset. Using simulated initial data, we compare our method with three advanced learning path recommendation methods, SRC, GEHRL, and DLPR, to demonstrate the effectiveness of our approach in online education settings. The results of the simulation experiments are shown in Figure 7. We observe that our method performs better than the other models on the simulated data. This indicates that, when facing with complex and dynamic student information, our method exhibits strong adaptability. It outperforms existing advanced learning path recommendation methods and hence being well-suited for application in online education scenarios.

These observations collectively demonstrate that our method generates more informative KC prerequisite relation graphs, addressing key limitations in the original datasets, such as missing or incorrect relations and incomplete concept coverage. Moreover, none of the datasets originally contain similarity relations, further highlighting the value of our approach in enriching the KC structure to support more effective LPR.

Running Time Analysis (RQ6) We evaluate the running time of the Edu-GraphRAG module and the DLRL module, as shown in Table 4. All runtime evaluations are conducted on a single NVIDIA 4090 GPU equipped with 256 GB of memory. We find that the Edu-GraphRAG module efficiently constructs the knowledge structure graph, completing the process within minutes. Although the DLRL module incurs a slightly higher runtime compared to DLPR methods (Zhang et al. 2024), this overhead is well justified by its significant performance gains.

Module	Junyi	MCX	ASS09
Edu-GRAG	12m 53s	9m 7s	5m 6s
DLRL	8h 23m 26s	2h 29m 48s	2h 1m 17s
DLPR	7h 37m 49s	2h 13m 15s	1h 40m 30s

Table 4: Running time for the Edu-GraphRAG (Edu-GRAG) module and the DLRL module.

τ	**0.1**	**0.01**	**0.001**	**0.0001**
Junyi	0.2157	0.2142	**0.2431**	0.2037
MCX	0.1478	0.2800	**0.3508**	0.1655
ASS09	0.0909	0.0829	**0.1174**	0.1028

Table 5: Performance with different threshold τ.

Analysis of Threshold τ (RQ7) We train KnowLP with various level thresholds τ on three benchmarks. As shown in Table 5, $\tau = 0.001$ consistently achieves the best performance and is thus used in all experiments.

Conclusion

In this paper, we propose a KnowLP method to achieve a more effective learning path by considering dual KC structures, prerequisite relations, and similarity relations. To be specific, we propose an EDU-GraphRAG module that adaptively generates graphs for KCs based on diverse educational datasets, enhancing the applicability of existing learning path recommendation methods. Then, we develop a DLRL module that alleviates the blocked phenomenon, further improving the performance of learning path recommendations. Finally, we conduct extensive experiments on three datasets, demonstrating that our method not only achieves state-of-the-art performance but also generates more effective and longer learning paths.

Acknowledgments

We would like to thank Guiyong Zhuang for helpful comments. This work supported by National Key R&D Program of China (No. 2022YFC3303603), National Natural Science Foundation of China (Nos. 62377028, 62477016 and 62276114), Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515120064 and 2024A1515140144), Guangzhou Science and Technology Planning Project (No. 2025A03J3565), Fundamental Research Funds for the Central Universities (No. 21625102).

References

Chen, C.-M. 2009. Ontology-Based Concept Map for Planning a Personalised Learning Path. *British Journal of Educational Technology*, 40: 1028–1058.

Chen, X.; Shen, J.; Xia, W.; Jin, J.; Song, Y.; Zhang, W.; Liu, W.; Zhu, M.; Tang, R.; Dong, K.; Xia, D.; and Yu, Y. 2023. Set-to-Sequence Ranking-based Concept-aware Learning Path Recommendation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI-2023)*, 5027–5035.

Chinda, W. 2022. Effect of Gagne's Learning Hierarchy on Chemistry Student Achievement in Senior Secondary School. *Iconic Research and Engineering Journals*, 5: 609–618.

Cover, T.; and Hart, P. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13: 21–27.

Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; and Jurišica, L. 2014. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering*, 96: 59–69.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; and Larson, J. 2024. From local to global: A graph Rag Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*.

Gagné, R. M. 1968. Presidential Address of Division 15 Learning Hierarchies. *Educational Psychologist*, 6: 1–9.

Goodwin, T. R.; Savery, M. E.; and Demner-Fushman, D. 2020. Flight of the PEGASUS? Comparing Transformers on Few-shot and Zero-shot Multi-document Abstractive Summarization. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING-2020)*, 5640.

Guan, Q.; Cheng, X.; Xiao, F.; Li, Z.; He, C.; Fang, L.; Chen, G.; Gong, Z.; and Luo, W. 2025. Explainable Exercise Recommendation with Knowledge Graph. *Neural Networks*, 183: 106954.

Hidasi, B. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939*.

Intayoad, W.; Kamyod, C.; and Temdee, P. 2020. Reinforcement Learning Based on Contextual Bandits for Personalized Online Learning Recommendation Systems. *Wireless Personal Communications*, 115: 2917–2932.

Konda, V.; and Tsitsiklis, J. 1999. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems (NIPS-1999)*.

Kubotani, Y.; Fukuhara, Y.; and Morishima, S. 2021. Rltutor: Reinforcement Learning Based Adaptive Tutoring System by Modeling Virtual Student with Fewer Interactions. *arXiv preprint arXiv:2108.00268*.

Laskar, M. T. R.; Hoque, E.; and Huang, J. 2020. Query Focused Abstractive Summarization via Incorporating Query Relevance and Transfer Learning with Transformer Models. In *Proceedings of the 33rd Canadian Conference on Artificial Intelligence (Canadian AI-2020)*, 342–348.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented Generation for Knowledge-intensive Nlp Tasks. In *Advances in Neural Information Processing Systems (NeurIPS-2020)*, 9459–9474.

Li, Q.; Xia, W.; Yin, L.; Jin, J.; and Yu, Y. 2024. Privileged Knowledge State Distillation for Reinforcement Learning-based Educational Path Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD-2024)*, 1621–1630.

Li, Q.; Xia, W.; Yin, L.; Shen, J.; Rui, R.; Zhang, W.; Chen, X.; Tang, R.; and Yu, Y. 2023. Graph Enhanced Hierarchical Reinforcement Learning for Goal-oriented Learning Path Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM-2023)*, 1318–1327.

Lin, C. F.; Yeh, Y.-c.; Hung, Y. H.; and Chang, R. I. 2013. Data Mining for Providing a Personalized Learning Path in Creativity: An Application of Decision Trees. *Computers & Education*, 68: 199–210.

Liu, Q.; Tong, S.; Liu, C.; Zhao, H.; Chen, E.; Ma, H.; and Wang, S. 2019. Exploiting Cognitive Structure for Adaptive Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD-2019)*, 627–635.

Ma, H.; Qi, Z.; Dong, X.; Li, X.; Zheng, Y.; Meng, X.; and Meng, L. 2023. Cross-modal Content Inference and Feature Enrichment for Cold-start Recommendation. In *International Joint Conference on Neural Networks (IJCNN-2023)*, 1–8.

Meng, L.; Qi, Z.; Wu, L.; Du, X.; Li, Z.; Cui, L.; and Meng, X. 2024. Improving Global Generalization and Local Personalization for Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36: 76–87.

Minn, S.; Zhu, F.; and Desmarais, M. C. 2018. Improving Knowledge Tracing Model by Integrating Problem Difficulty. In *Proceedings of 18th IEEE International Conference on Data Mining Workshop on PhD Forum*, 1505–1506.

Nabizadeh, A. H.; Goncalves, D.; Gama, S.; Jorge, J.; and Rafsanjani, H. N. 2020. Adaptive Learning Path Recommender Approach Using Auxiliary Learning Objects. *Computers & Education*, 147: 103777–103793.

Nabizadeh, A. H.; Mário Jorge, A.; and Paulo Leal, J. 2017. Rutico: Recommending Successful Learning Paths under Time Constraints. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP-2017)*, 153–158.

Pan, S. C.; Selvarajan, G.; and Murphy, C. S. 2024. Interleaved Pretesting Enhances Category Learning and Classification Skills. *Journal of Applied Research in Memory and Cognition*, 13: 393–406.

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

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-Context

Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Rohrer, D. 2012. Interleaving Helps Students Distinguish Among Similar Concepts. *Educational Psychology Review*, 24: 355–367.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shen, S.; Huang, Z.; Liu, Q.; Su, Y.; Wang, S.; and Chen, E. 2022. Assessing Student's Dynamic Knowledge State by Exploring the Question Difficulty Effect. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2022)*, 427–437.

Shi, D.; Wang, T.; Xing, H.; and Xu, H. 2020. A learning Path Recommendation Model Based on a Multidimensional Knowledge Graph Framework for E-learning. *Knowledge-Based Systems*, 195: 105618–105628.

Wang, H.; Long, T.; Yin, L.; Zhang, W.; Xia, W.; Hong, Q.; Xia, D.; Tang, R.; and Yu, Y. 2023a. GMOCAT: A Graph-Enhanced Multi-Objective Method for Computerized Adaptive Testing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD-2023)*, 2279–2289.

Wang, J.; Cui, Z.; Wang, B.; Pan, S.; Gao, J.; Yin, B.; and Gao, W. 2024a. IME: Integrating Multi-curvature Shared and Specific Embedding for Temporal Knowledge Graph Completion. In *Proceedings of the ACM Web Conference (WWW-2024)*, 1954–1962.

Wang, J.; Kai, S.; Luo, L.; Wei, W.; Hu, Y.; Liew, A. W.-C.; Pan, S.; and Yin, B. 2024b. Large Language Models-guided Dynamic Adaptation for Temporal Knowledge Graph Reasoning. *Advances in Neural Information Processing Systems*, 37: 8384–8410.

Wang, J.; Wang, B.; Qiu, M.; Pan, S.; Xiong, B.; Liu, H.; Luo, L.; Liu, T.; Hu, Y.; Yin, B.; and Gao, W. 2023b. A Survey on Temporal Knowledge Graph Completion: Taxonomy, Progress, and Prospects. *arXiv preprint arXiv:2308.02457*.

Xu, D.; Wang, Z.; Chen, K.; and Huang, W. 2012. Personalized Learning Path Recommender Based on User Profile Using Social Tags. In *Proceedings of the 5th International Symposium on Computational Intelligence and Design (ISCID-2012)*, 511–514.

Yang, X.; Yan, M.; Pan, S.; Ye, X.; and Fan, D. 2023. Simple and Efficient Heterogeneous Graph Neural Network. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI-2023)*, 10816–10824.

Yuksekgonul, M.; Bianchi, F.; Boen, J.; Liu, S.; Lu, P.; Huang, Z.; Guestrin, C.; and Zou, J. 2025. Optimizing Generative AI by Backpropagating Language Model Feedback. *Nature*, 639: 609–616.

Zhang, H.; Bu, C.; Liu, F.; Liu, S.; Zhang, Y.; and Hu, X. 2022. APGKT: Exploiting Associative Path on Skills Graph for Knowledge Tracing. In *Proceedings of 19th Pacific Rim International Conference on Artificial Intelligence (PRICAI-2022)*, 353–365.

Zhang, H.; Shen, S.; Xu, B.; Huang, Z.; Wu, J.; Sha, J.; and Wang, S. 2024. Item-Difficulty-Aware Learning Path Recommendation: From a Real Walking Perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD-2024)*, 4167–4178.

Zhang, H.; Wu, B.; Yang, X.; Yuan, X.; Liu, X.; and Yi, X. 2025. Dynamic Graph Unlearning: A General and Efficient Post-Processing Method via Gradient Transformation. In *Proceedings of the ACM on Web Conference (WWW-2025)*, 931–944.

Zhang, H.; Yuan, X.; and Pan, S. 2024. Unraveling Privacy Risks of Individual Fairness in Graph Neural Networks. In *IEEE 40th International Conference on Data Engineering (ICDE-2024)*, 1712–1725.

Zhang, Q.; Lu, J.; and Zhang, G. 2021. Recommender Systems in E-learning. *Journal of Smart Environments and Green Computing*, 1: 76–89.

Zheng, C.; Fan, X.; Pan, S.; Jin, H.; Peng, Z.; Wu, Z.; Wang, C.; and Philip, S. Y. 2023. Spatio-Temporal Joint Graph Convolutional Networks for Traffic Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36: 372–385.