Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-025-10808-4

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

KE-LPG: Toward Semantic Refinement for Lesson Plan Generation

Hanghui Guo[1] · Jia Zhu[2] · Changfan Pan[1] · Qing Wang[1] · Cong Zhou[1] · Chaojun Meng[1]

Received: 22 April 2024 / Accepted: 23 October 2025

Abstract

Lesson plan design can optimize the teaching process, making teaching more targeted and scientific, thereby improving teaching quality and student learning outcomes. But teachers often struggle to complete high-quality lesson plan design due to limited teaching time and insufficient design skills. To solve this problem, the emergence of large language models (LLMs) provides teachers with a more efficient way to plan teaching content, design teaching activities, and optimize teaching strategies, thus enhancing the quality and effectiveness of lesson plan design. However, existing LLMs have limitations. Their static nature leads to a lack of professional education knowledge and an inability to continuously learn new knowledge. As a result, they perform poorly when dealing with unfamiliar content and struggle to generate detailed and high-quality lesson plans. To address these issues, we propose a Knowledge-Enhanced Lesson Plan Generation method (KE-LPG). Specifically, this method extracts subgraphs related to lesson plan content from a subject-specific knowledge graph. By combining techniques like graph Laplacian learning and semantic relevance calculation, it accurately constructs a keyword graph to enhance the LLM's knowledge expression and understanding in the education field. Furthermore, during the pre-training stage, the constructed keyword graph is integrated with lesson plan content for fine-tuning, improving the LLM's performance in lesson plan generation. To our knowledge, this is the first time that a semantic refinement approach has been used to generate lesson plans. Experimental results show that our method not only provides high-quality curriculum plans but also ensures the reliability of the generated knowledge points. The resource of the paper is available at https://github.com/ghh1125/data.

Keywords Large language model · Knowledge graph · Lesson plan · Educational technology · Personalized instruction

1 Introduction

Lesson plan development serves as a fundamental component of educational instruction, systematically organizing teaching objectives, content, methodologies, and assessment strategies to enhance the precision and scientific rigor of the educational process, thereby effectively improving teaching quality and student learning outcomes [1–3]. The importance of structured lesson planning is well-grounded in established learning theories, such as Bloom's Taxonomy, which classifies educational objectives into cognitive levels-remembering, understanding, applying, analyzing, evaluating, and creating—providing a scaffold for curriculum design and assessment alignment [4, 5]. However, creating personalized lesson plans requires navigating a complex process that must accommodate students' diverse backgrounds and learning capabilities, significantly increasing the complexity during the design phase [6, 7]. Educators frequently encounter constraints of limited resources and time pressures,

✉ Hanghui Guo
ghh1125@zjnu.edu.cn

Jia Zhu
jiazhu@zjnu.edu.cn

Changfan Pan
changfanpan@zjnu.edu.cn

Qing Wang
wq2481@zjnu.edu.cn

Cong Zhou
zhoucong@zjnu.edu.cn

Chaojun Meng
mengchaojun@zjnu.edu.cn

[1] School of Computer Science and Technology, Zhejiang Normal University, 688 Yingbin Avenue, Jinhua 321004, Zhejiang, China

[2] Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, 688 Yingbin Avenue, Jinhua 321004, Zhejiang, China

Published online: 11 November 2025

making it challenging to develop specialized, high-quality lesson plans within compressed timeframes [8, 9].

With the advancement of deep learning technologies, language models have achieved remarkable breakthroughs, particularly through the rapid development of generative artificial intelligence technologies [10, 11]. The emergence of the transformer architecture has spawned numerous large language models (LLMs) based on encoder and decoder frameworks. Notably, decoder-based GPT has emerged as a milestone in this field [12]. These dual-stage generative GPT models exhibit exceptional generalization capabilities, enabling them to adeptly address various natural language processing tasks. By leveraging the strengths of LLMs, education can provide more accessible and personalized services for both educators and students [13].

Despite considerable achievements in natural language processing, LLMs still face certain limitations in comprehending real-world contexts [14, 15]. Particularly in the educational domain, LLMs lack specialized expertize in pedagogical practices, making it difficult to generate high-quality lesson plans that could potentially impact student learning negatively [16]. Furthermore, existing hallucination issues and the absence of domain-specific knowledge cannot guarantee content authenticity and reliability [15, 17].

Knowledge graphs (KGs) can effectively represent diverse entities and relationships, establishing comprehensive knowledge representation frameworks that clarify and streamline knowledge relationships [18–20]. This structured knowledge representation enables machine learning algorithms to better interpret and apply knowledge more effectively [21, 22]. However, knowledge representation in KGs is typically static, failing to capture dynamic changes and complex contexts of knowledge, thereby limiting the model's ability to comprehend real-world scenarios [23–25]. In the educational field, despite the substantial advantages that KGs provide in encapsulating various entities and interactions, the process of creating comprehensive lesson plans solely from KGs presents multiple significant challenges [26]. Lesson plans encompass not only numerous hierarchical relationships among courses, teaching objectives, resources, and activities, but also rich semantic information including detailed descriptions of instructional resources and articulations of learning objectives. The advantages and limitations of the existing method and our method KE-LPG are shown in Table 1.

To address the respective limitations of LLMs and KGs in educational applications, our research proposes an innovative approach called KE-LPG (Knowledge-Enhanced Lesson Plan Generation) that combines the capabilities of both KGs and LLMs to tackle key challenges in lesson plan generation. The primary challenge of this technology lies in effectively integrating external information sources with language models to ensure that the resulting educational content maintains high quality and accuracy. We present an innovative methodology that employs dependency matrix algorithms to extract keywords and construct keyword graphs, effectively utilizing KGs while enhancing the expressive capabilities of language models. Additionally, to enable language models to better absorb and comprehend essential details within KGs, we utilize Graph Convolutional Networks (GCNs) to process and analyze KGs, generating tags suitable for language model input sequences. Through fine-tuning and training LLMs, we ensure that the generated lesson plans achieve optimal quality and personalization levels.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the pioneers in mining the uniqueness of lesson plan data and combining it with LLM to improve lesson plan generation. This effort helps substantially advance interdisciplinary applications.
- We introduced KE-LPG, a new method that combines KG and LLM, using GCN and knowledge-enhanced methods to capture information and improve LLM's ability to create effective and high-quality lesson plans. This innovative method amplifies the reasoning capabilities of the expansive language model and elevates its interpretability, surpassing the performance achieved by relying solely on the prowess of the LLM.
- We conducted experimental evaluations of this method and compared it to other LLM with larger parameters. The experimental results show that using the KE-LPG method has significantly improved the evaluation indicators of detailed lesson plan generation. The text readability score increased by 2%. Expert ratings show that scores improved by about 5% to 6%.

The remainder of this article is structured as follows: Section 2 provides a brief overview of related work concerning the integration of Knowledge Graphs and Large Language Models, as well as the semantic refinement of lesson plans. In Section 3, we delve into the technical intricacies of our method. Section 4 outlines our experimental findings. Lastly. Section 5 provides a discussion and limitations of this article. Section 6 concludes the full paper and discusses possible future work directions.

2 Related Work

We investigate three domains: integration of KG and LLM, lesson plan generation, and semantic refinement.

Table 1 Comparative analysis of existing approaches for lesson plan generation

Approach category	Representative works	Main contributions	Advantages	Limitations
Template-based systems	Rule-based educational frameworks, Taxonomy-driven generators	Structured lesson plan templates, Educational standard compliance	• Clear structure	• Limited flexibility
			• Standards alignment	• Generic content
			• Predictable output	• No personalization
			• Manual template creation	
Pure LLM approaches	GPT-based lesson generators, BERT-enhanced educational content	Natural language generation, Context-aware content creation	• Fluent text generation	• Domain knowledge gaps
			• Context understanding	• Hallucination issues
			• Scalable deployment	• Lack of factual reliability
			• No structured knowledge	
Knowledge graph methods	Educational ontology systems, Concept map generators	Structured knowledge representation, Relationship modeling	•Rich semantic relationships	• Static representations
			• Domain expertize	• Limited text generation
			• Factual accuracy	• Poor content coherence
			• Logical consistency	• Scalability challenges
Our KE-LPG method	Knowledge-Enhanced Lesson Plan Generation	Advanced semantic integration, GCN-based knowledge processing, Dependency matrix algorithms	• Deep knowledge integration	• Computational complexity
			• Semantic-aware generation	• Training data dependency
			• Personalization support	
			• Quality assurance	
			• Dynamic knowledge utilization	

2.1 Integration of KG and LLM

Large language models such as BERT [27–29], RoBERTA [30], and T5 [31], along with subsequent advancements such as GPT3 [32, 33], GPT-4 [34, 35] and LLaMA [36], have demonstrated remarkable capabilities. However, challenges like hallucinations in large language models [37, 38] have prompted a closer examination of these models. To solve these problems, people are combining KG with LLM. Incorporating KG into these models offers more meaningful, insightful, and trustworthy explanations [39, 40]. In the realm of LLM, predicting observed phenomena becomes implicit [41]. KG stores large amounts of information in an explicit and structured manner, augmenting the knowledge awareness of large predictive models [42]. Various methods have been proposed to exploit this synergy. Enhanced language representation models like ERNIE [43] are trained using large-scale text corpora and KG, simultaneously incorporating vocabulary, syntax, and knowledge information. Other approaches, such as the SKEP method [44], utilize KG to enhance pre-training. In recent years, innovative methods have emerged, incorporating KG into LLM during pre-training. Examples include GNN [45], QA-GNN [46], FDGNN [47], KALA [48], and DKPLM [49]. This integration allows LLM to acquire knowledge more effectively, facilitating improved fine-tuning. Furthermore, KG is inte-

grated into the input of LLM, such as in K-BERT [50]. In the first step, knowledge triplets are injected into sentences through a visible matrix, where only knowledge entities can access the knowledge triplet information, while the tokens in the sentences can only see each other in the self-attention module. To further reduce knowledge noise, Colake [51] proposes a unified word-knowledge graph, where tokens in the input sentences form a fully connected word graph, with tokens aligned with knowledge entities connected to their adjacent entities. The relationship between LLM and KG is anticipated to address challenges related to content accuracy and ethical considerations [43, 52].

2.2 Lesson Plan Generation

Relevant work in the field of lesson plan generation [53] encompasses diverse research and applications. In recent years, the integration of advanced technologies like natural language processing (NLP) and machine learning (ML) has influenced automated education tools. These tools are designed to assist educators in developing more effective and personalized lesson plans based on the diverse needs and teaching goals of students [54, 55]. There is a concerted effort in developing instructional design systems dedicated to creating tools capable of generating lesson plans based on subjects and learning stages [56, 57]. These systems match the rules in education and what we are supposed to teach. Intelligent education systems have emerged to analyze student learning data [58]. By doing so, they adapt and generate personalized plans, catering to the varied learning styles of different students. Research on cognitive psychology and instructional design principles has influenced the development of automated lesson plan generation systems, ensuring alignment with pedagogical theories [59]. In particular, Bloom's Taxonomy has been widely used as a theoretical foundation for structuring learning objectives in lesson plans. Its hierarchical model of cognitive skills-from remembering to creating-offers a valuable framework for aligning instructional goals with assessment and content delivery strategies [4, 5, 60, 61]. Several systems have begun incorporating such taxonomies to guide content selection and task complexity, ensuring that generated lesson plans not only meet curricular standards but also support cognitive development at appropriate levels. Some studies have explored the use of reinforcement learning algorithms to optimize the generation of personalized lesson plans, considering factors such as student engagement and knowledge retention [62]. Another research focuses on leveraging large-scale educational data for analysis [63, 64]. They want to find the best ways to teach and put them into lesson plans that get made automatically. Furthermore, collaborative course design tools have been developed to facilitate knowledge sharing and collaboration among teachers, ultimately enhancing the overall quality of courses [65]. The ongoing efforts related to lesson plan generation aim to empower educators with additional tools and resources [66], providing valuable support in addressing the diverse needs of students in their daily teaching practices.

2.3 Semantic Refinement

The concept of semantic refinement is deliberated in related fields, and it pertains to enhancing the semantic quality of generated outcomes through a more profound understanding and optimization of information [67–69]. For instance, in semantic segmentation, the two-stage refinement network, DRNet [70] and ISL [71], contribute to improved semantic segmentation performance, thereby advancing semantic refinement. Through the processing of semantics, the system gains a heightened ability to comprehend and generate intricate language structures [72–74]. This approach analyzes semantic relationships, contextual information, and entity recognition to acquire more precise and comprehensive semantic expressions [75]. Semantic refinement has broad application prospects in enhancing model performance, optimizing information retrieval, and refining generation tasks. Additionally, research has proposed the MCRNet [76] for semantic segmentation, aiming to enhance the model's capability to capture spatial and contextual information. By extracting image features at multiple stages, MCRNet achieves superior segmentation performance compared to traditional methods. Refign is also a good approach for aligning and refining semantic segmentation to accommodate adverse conditions [77]. In summary, semantic refinement plays a crucial role in enhancing model performance and optimizing information processing tasks.

2.4 Summary

However, a unified framework that simultaneously leverages deep semantic integration, structured knowledge, and pedagogical alignment for high-quality, personalized lesson plan generation remains underexplored. Our proposed KE-LPG method aims to bridge this gap by combining KG-augmented LLMs with semantic-aware generation and cognitive theory-driven structuring.

3 Methods

In this section, we present our KE-LPG method with technical details. As shown in Fig. 1, this method contains two key stages of KE-LPG: enhancing expression ability and enhancing semantic refinement.

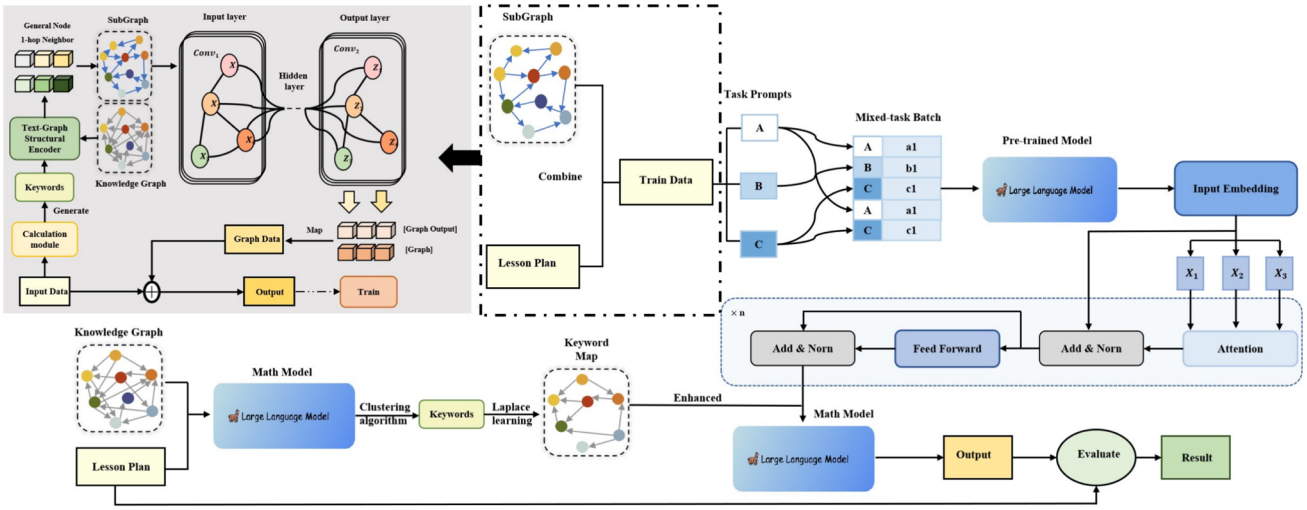

Fig. 1 Specific working diagrams for lesson plan generation. We use KE-LPG before generation, which is mainly divided into two aspects: improving the semantic refinement of lesson plan generation by enhancing model input and using KG to enhance the expressive ability of the model

3.1 Enhancing Expression Ability

In this subsection, we first introduce how to enhance the expressive capabilities of large language models (LLMs) through the use of knowledge graphs (KGs). As illustrated in Fig. 2, KE-LPG leverages an LLM to extract keywords from the knowledge base and then evaluates the relationship weights between these keywords using a graph Laplacian learning approach based on a dependency matrix algorithm. This innovative method combines vector similarity with graph-based associations, enabling the LLM to effectively utilize the information contained in the KG. Our dependency matrix algorithm consists of two key components: keyword extraction and relationship construction.

The keyword extraction process begins with the use of a pre-trained LLM to encode textual data obtained from the KG. These LLMs are adept at capturing intricate semantic relationships and contextual variations present within the text. By leveraging such models, the textual information is transformed into high-dimensional vector representations, thereby preserving semantic coherence. Following the encoding stage, the resulting vectors are subjected to unsupervised clustering methodologies. These algorithms partition the textual input into cohesive clusters based on semantic similarity. Through this process, semantically related words and phrases are grouped together, forming distinct clusters that represent the underlying thematic structures within the KG. Representative keywords are then extracted from each cluster.

To ensure the quality and relevance of the extracted keywords, a series of refinement steps is undertaken. This refinement process includes the elimination of stop words, which are commonly occurring but lack semantic significance, and the exclusion of terms that appear with either excessively low or high frequency in the text. These filtering mechanisms aim to enhance the precision of the keyword list, ultimately producing a refined set of keywords that encapsulates the substantive content and educational breadth of the KG.

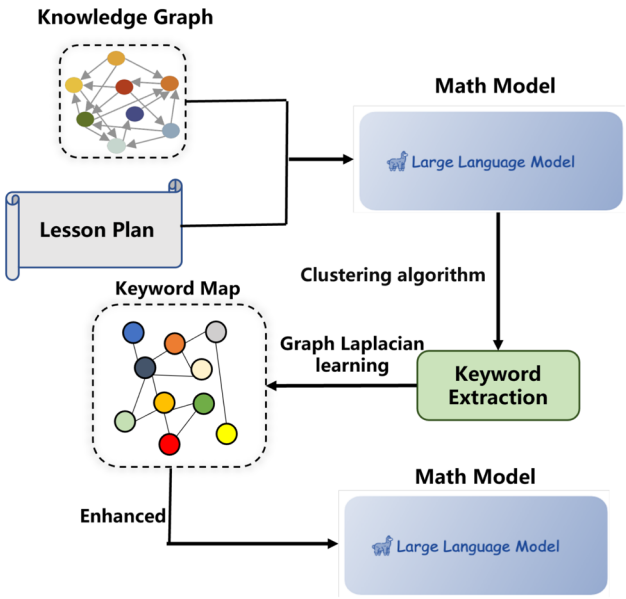

Fig. 2 Schematic diagram illustrating keyword graph enhancement via graph Laplacian learning

Generate high-quality keyword lists by encoding, clustering, extracting keywords from text data, filtering and refining them. The clustering process is as follows:

$$L = \prod_{j=1}^{n} \sum_{i=1}^{k} \pi_i \mathcal{N}(v_j | \mu_i, \Sigma_i), \tag{1}$$

where π_i is the weight of the i-th Gaussian distribution, subject to $\sum_{i=1}^{k} \pi_i = 1$, and $\mathcal{N}(v_j|\mu_i, \Sigma_i)$ represents the probability density function of the i-th Gaussian distribution at point v_j.

$$w_{ij} = \frac{\pi_i \mathcal{N}(v_j|\mu_i, \Sigma_i)}{\sum_{l=1}^{k} \pi_l \mathcal{N}(v_j|\mu_l, \Sigma_l)}, \tag{2}$$

where w_{ij} denotes the posterior probability that data point v_j belongs to the i-th Gaussian distribution.

$$\mu_i = \frac{\sum_{j=1}^{n} w_{ij} v_j}{\sum_{j=1}^{n} w_{ij}}, \tag{3}$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} w_{ij} (v_j - \mu_i)(v_j - \mu_i)^T}{\sum_{j=1}^{n} w_{ij}}, \tag{4}$$

$$\pi_i = \frac{\sum_{j=1}^{n} w_{ij}}{n}, \tag{5}$$

where μ_i is the mean vector of the i-th Gaussian distribution, Σ_i is its covariance matrix, and π_i is its weight.

The Extract function selects representative keywords K_i from each cluster C_i. A common approach is to choose the word closest to the mean of the cluster as the representative keyword. Mathematically, this can be represented as:

$$K_i = \arg\min_{v \in C_i} \|v - \mu_i\|, \tag{6}$$

where K_i represents the representative keyword of cluster C_i, μ_i is the mean vector of cluster C_i, and $\|\cdot\|$ denotes the Euclidean distance.

In the relationship construction stage, connections between keywords are established through the creation of a keyword graph, using co-occurrence frequencies as the foundation for determining relationships. In this graph, edges represent the frequency with which keyword pairs co-occur, and the weight of each edge reflects the strength of their co-occurrence. A critical component of this process is the role of the LLM, particularly in the extraction and refinement of keywords.

To construct these relationships, co-occurrence matrices are employed as essential tools for capturing the associations among keywords. These matrices provide a structured representation of how frequently different terms appear together within the textual data. Once the co-occurrence matrices are constructed, semantic similarities between keywords are evaluated using graph Laplacian learning methods. This technique allows the keyword graph to reveal underlying patterns of similarity and dissimilarity among terms.

By applying this computational approach, the system can extract meaningful insights from the keyword graph, facilitating the identification of closely related concepts within the semantic space. Ultimately, this process enables the discovery of complex semantic relationships embedded in the text, enriching the overall understanding of the educational knowledge domain.

The similarity between keywords w_i and w_j is determined by the Semantic association measure (Sam) formula:

$$\text{Sam}(w_i, w_j) = \frac{\text{Aff}(w_i, w_j)}{\sqrt{\text{Pot}(w_i) \cdot \text{Pot}(w_j)}}, \tag{7}$$

where $\text{Aff}(w_i, w_j)$ denotes the affinity between terms w_i and w_j, while $\text{Pot}(w_i)$ represents the potency of term w_i.

Our method differs from traditional methods of leveraging KGs to support LLMs by employing a dependency matrix algorithm that emphasizes keywords extracted from lesson plans and their interrelationships within the educational KG. This strategy enhances the LLM's ability to comprehend and utilize the rich, domain-specific information present in the education field.

3.2 Enhancing Semantic Refinement

This subsection is primarily divided into two key components: data and KG processing, and training input. The first component focuses on ensuring a high-quality foundation for model training through precise handling of both data and the knowledge graph. The second component emphasizes the integration of this data with subgraphs processed by graph convolutional networks (GCNs), and how to effectively input this combined information into the model for training. Together, these components form the core of this stage, which aims to generate lesson plans with richer and more detailed semantics. The specific process is illustrated in Fig. 3.

3.2.1 Data and KG Processing

In the data preprocessing phase, we ensure the quality of training data, particularly within the context of educational content. To achieve this, we employ a series of preprocessing techniques designed to filter out noise, resolve ambiguities, and efficiently extract valuable insights.

Initially, the textual data is transformed into a specialized key-value pair format. Next, we apply Term Frequency-Inverse Document Frequency (TF-IDF) analysis across the entire dataset to identify critical keywords. These extracted keywords are then integrated with the existing KG to construct a subgraph that encapsulates the relevant concepts.

To enhance the granularity of node representation, we expanded the dimensions of the graph convolutional network (GCN) layers to configurations of 1×16 and 16×1. This architectural adjustment enables a more detailed exploration of the KG's topological structure and captures subtle variations in the interactions between nodes. Moreover, the iterative refinement process plays a pivotal role in improv-

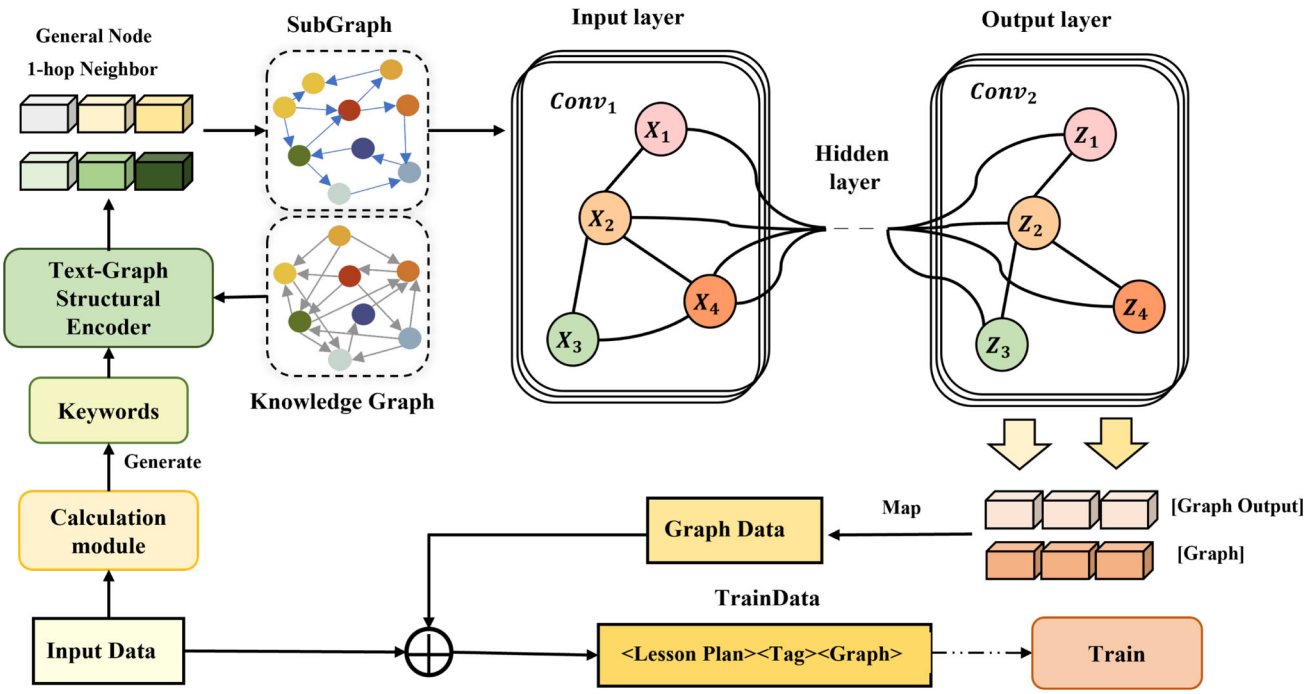

Fig. 3 Enhanced semantic refinement technology framework

ing the quality of educational lesson plans. By continuously revisiting and refining the generated subgraph, we aim to fine-tune the representation of key concepts and their interrelationships. This iterative enhancement is essential for aligning the output with pedagogical goals and effectively addressing the learning needs of the target audience.

Our objective is to update the node features using a GCN. The update equation of GCN is as follows: let $\mathbf{X}$ denote the input feature matrix representing the features of nodes. $\mathbf{X}$ has dimensions $N \times D$, where N is the number of nodes and D is the feature dimension. We use the adjacency matrix $\mathbf{A}$ to represent the graph's connectivity structure, where $\mathbf{A}$ also has dimensions $N \times N$. $\mathbf{A}_{ij} = 1$ indicates there is an edge between node i and node j, otherwise it's 0.

$$\mathbf{H}^{(l+1)} = f\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}^{(l)}\right), \tag{8}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with added self-loops, $\tilde{\mathbf{D}} = \text{diag}\left(\sum_j \tilde{\mathbf{A}}_{ij}\right)$ is the diagonal degree matrix, $\mathbf{X}$ represents the node feature matrix, $\mathbf{W}^{(l)}$ represents the weight matrix at the l-th layer, $\mathbf{I}$ is the identity matrix, and diag$(\cdot)$ denotes taking the diagonal elements. This completes the computation process of a full graph convolutional layer.

Through the above steps, the learning of local information is used to enhance the understanding and reasoning ability of the entire KG, thereby improving the generalization ability and expression ability of the model.

3.2.2 Training Input Data

In this section, we detail the training input process, with a focus on integrating the preprocessed data and the subgraph generated by the graph convolutional network (GCN). The preparatory phase involves aligning the input data with the model's architectural requirements. Fundamental preprocessing steps-including tokenization, normalization, and encoding-are employed to ensure the data is in a format compatible with neural network ingestion. These steps standardize the input, preserve semantic integrity, and facilitate efficient processing during training.

The subgraph generated by the GCN captures complex relationships among keywords extracted from the educational content. By leveraging graph-based representations, the subgraph enriches the input data with semantic connections and contextual information derived from the broader knowledge graph (KG). However, effectively enabling the large language model (LLM) to utilize the key information embedded in the GCN-generated subgraph to guide lesson plan generation presents a significant challenge.

To address this, we introduce a labeling strategy that helps the model distinguish between textual input and subgraph-derived information. These labels act as essential markers, directing the model on how to incorporate the subgraph's knowledge during training. This labeling technique enables precise annotation of the input data, allowing the model to apply subgraph information in a targeted manner to enrich and optimize lesson plan generation. As a result, the model's

ability to interpret and utilize the KG is enhanced, leading to improved comprehension and application of educational content, and ultimately elevating the model's overall performance.

We integrate the subgraph's features into the LLM's training process through encoding and prompt-based instruction. These prompts explicitly guide the model to leverage the subgraph's semantic information when generating lesson plans. This strategy not only results in higher-quality, semantically rich outputs but also improves the model's performance in understanding and applying domain-specific knowledge.

3.3 Summary of Methods

By following these two primary stages, we gain deeper insights into the relevant knowledge within the educational domain during LLM training. This process facilitates the generation of lesson plans that are not only more semantically rich but also better tailored to individual learning needs.

4 Experiments

This section presents our extensive experiments, validating the effectiveness of our approach through a range of evaluation metrics and expert assessments.

4.1 Experiment Datasets and Evaluation Metrics

Our choose datasets from students' internships at primary and secondary educational over three years. These datasets include internship experiences, training data for scoring, and a curated subset focusing on mathematics subjects. We also build an education KG to enrich our model's training. This KG aids in developing a deep understanding of mathematical principles, enhancing the generation of lesson plans.

For evaluating our model's performance, we define diverse metrics including perplexity, Rouge, and BLEU scores, alongside metrics tailored for Chinese text. These metrics provide effective tools for comparing and optimizing the generated lesson plans, ensuring comprehensive assessment and optimization of our method.

(1) Educational-effectiveness. This is a core metric in our evaluation, designed to measure the pedagogical soundness of generated lesson plans. It is composed of four dimensions, consistent with the educator ratings shown in Table 4:
Preliminary analysis (Ea): assessing learner analysis, content analysis, and environmental analysis to ensure comprehensive and accurate preparation.
Teaching objectives (To): evaluating the clarity, alignment with curriculum standards, and hierarchical structure of stated learning objectives.
Teaching process design (Tpd): assessing logical coherence, methodological diversity, quality of interaction, and timeliness of feedback.
Documentation specification (Ds): evaluating structural organization, formatting clarity, and linguistic accuracy of the lesson plan.
Each dimension is scored individually and aggregated into a composite score, which is then normalized into the range [0, 25]. Higher scores reflect stronger educational effectiveness.

(2) Minedit-similarity. This metric is derived from the minimum edit distance between the generated text and the reference lesson plan. We normalize the edit distance by the maximum string length, so the final score ranges from 0 (completely different) to 1 (identical). Lower raw edit distances or higher normalized similarity values represent better alignment with the reference. Our implementation follows the formulation used in prior text similarity studies.

(3) Readability metrics. We adopt Kincaid-grade, Kincaid-score, and SMOG-index, which are classical readability formulas originally developed for English texts. In our work, we use the Chinese-adapted implementations provided in the *cntext* library, which extend these formulas to Chinese by computing lexical and syntactic complexity based on character segmentation and sentence structure. Despite known limitations, these metrics provide an approximate measure of text difficulty in Chinese.

(4) Oread. This is an overall readability index provided by the cntext library. It integrates multiple linguistic features, including lexical diversity, syntactic depth, and cohesion indicators, to yield a composite score of text readability. Higher Oread values indicate clearer, more fluent, and more accessible text.

All custom metrics and implementations are cited from the cntext library (https://github.com/zhw3051/cntext), and we provide further justification for their use in the Discussion section. Oread is derived from two sub-indicators:

Lexical complexity (Lc): measured as the average number of characters per clause.

Syntactic complexity (Sc): measured as the proportion of adverbs and conjunctions within each sentence.

The final Oread score is computed as:

$$\text{Oread} = 100 - (Lc + Sc)/2. \tag{9}$$

4.2 Experiment Setup

Our research encompasses extensive experiments involving diverse models, including GPT3.5 [32], GPT2 [32], Baichuan2-13B [78], and ChatGLM2-6B [79]. And the KELPG method is integrated into the Baichuan and ChatGLM

Table 2 Experimental data on Chinese text readability

Model	Lc	Sc	Oread
Raw Data	59.22211	2.23584	69.27102
ChatGLM2-6B	56.95826	1.60578	70.71798
KE-LPG(ChatGLM)	51.79762	2.57937	**72.81151**
GPT2	139.8	2.2	29.0
GPT3.5 (135B)	37.96911	1.66544	78.18272
Baichuan2-13B	42.39827	2.20779	73.19697
KE-LPG(Baichuan)	39.29845	1.85614	**76.02483**

The bold values indicates the result value

models for enhanced training. We used an NVIDIA RTX A6000 for experiments. Evaluations included many metrics ,expert evaluation from mathematics teachers was also sought, focusing on four dimensions: preliminary analysis, teaching goal clarification, teaching process design, and document specification, each carrying a weighted score of 25%.

4.3 Experimental Results

We conducted experiments to evaluate the effectiveness of our KE-LPG method, assessing factors like text readability, similarity, and expert ratings.

4.3.1 Text Readability

Table 2 provides an overview of fine-tuned lesson plans across different models, evaluated using the cntext library's readability function with indicators Lc (Lexical Complexity), Sc (Syntactic Complexity), and Oread(Overall text readability). Original content scores approximately 69.27 without processing. Fine-tuning with GPT2, which has limited Chinese understanding, resulted in unreadable text. Direct use of ChatGLM-6B improved readability compared to the original text. Incorporating KG during ChatGLM2-6B training increased readability by approximately 2%. Despite reducing vocabulary complexity, sentence complexity moderately increased. The text readability achieved by the 6B model combined with the KG we use is close to the BaiChuan-13B model level, and we combine the KE-LPG method into the Baichuan model. Its readability is less than 2% different from GPT, but GPT3.5 is The 135B model is about 10 times larger than the model we used. This verifies the effectiveness of our method in enhancing the readability of generated text, demonstrating its academic value in improving text quality.

4.3.2 Comparative Analysis: KE-LPG versus ChatGLM2-6B

Table 3 presents a comparative analysis between ChatGLM2-6B and KE-LPG(ChatGLM), evaluating their performance across various metrics. KE-LPG outperforms ChatGLM2-6B in BLEU-4 scores, achieving 12.55231 compared to 10.40894, indicating higher lexical matching with the reference text. Additionally, KE-LPG demonstrates superior performance in metrics such as Kincaid level and score, Rouge-1, Rouge-2, Rouge-L, and SMOG, with notable improvements in Rouge indicators. Combining Kincaid-grade with KG resulted in lower scores, suggesting better suitability for students in lower grades. Furthermore, the KE-LPG method exhibits higher Jaccard similarity and lower minedit similarity compared to KG integrated lesson plans, indicating stronger consistency and requiring fewer editing operations for conversion. These findings highlight the enhanced learning potential and stronger consistency of text produced by the KE-LPG method.

Table 3 Comparative Analysis: KE-LPG vs ChatGLM2

Method	ChatGLM2	KE-LPG(ChatGLM2)
Bleu-4	10.40894	**12.55231**
Cosine-similarity	0.03002	**0.02128**
Educational-effectiveness	15.56189	**17.56284**
Kincaid-grade	38.80294	**20.06125**
Kincaid-score	10.32457	**58.24919**
Jaccard similarity	0.27021	**0.22145**
Minedit-similarity	2411.94044	**2389.47239**
Rouge-1	22.82476	**17.19830**
Rouge-2	9.82228	**6.98531**
Rouge-l	16.03698	**11.97214**
Smog-index	1.67086	**0.94051**

The bold values indicates the result value

It is worth noting that the evaluation metrics in Table 3 exhibit seemingly conflicting trends. For example, BLEU-4 improves while ROUGE-1/2/L decreases. This divergence can be explained by the different focuses of the metrics: BLEU emphasizes precise n-gram matches with the reference, while ROUGE prioritizes recall and coverage of reference expressions. Since KE-LPG introduces more domain-specific and semantically enriched expressions, exact matches increase (higher BLEU), but lexical overlap with the reference can decrease (lower ROUGE). Similarly, cosine and Jaccard similarities capture lexical overlap at the global level, which may drop if the generated text uses alternative but pedagogically valid phrasing. However, BLEU still recognizes the correctness of local matches.

Regarding readability metrics, the trends among Kincaid-grade, Kincaid-score, and SMOG appear inconsistent. We note that Kincaid-grade, Kincaid-score, and SMOG are originally designed for English, and their direct application to Chinese is not equivalent. In Table 6, the metrics are computed via the cntext library, which provides Chinese-adapted versions. Despite differences in sensitivities to lexical and syntactic complexity, the overall readability indicator Oread

Table 4 Professional Educator Ratings, evaluate from four dimensions: Early analysis (Ea), Teaching objectives (To), Teaching process design (Tpd), Documentation specifications (Ds)

Method	ChatGLM2							KE-LPG (ChatGLM)						
	A	B	C	D	E	F	G	A	B	C	D	E	F	G
Ea	17.3	15.1	18.2	19.4	16.9	15.5	20.1	19.2	18.1	21.5	22.0	19.1	18.2	23.2
To	20.1	21.4	18.4	18.2	19.6	17.4	18.1	21.1	21.1	21.1	20.4	20.0	20.1	18.4
Tpd	18.2	19.3	18.2	19.2	16.2	17.3	17.3	19.3	20.4	20.3	21.9	18.1	18.4	20.1
Ds	17.2	18.3	17.5	16.7	19.1	20.7	19.2	18.5	20.3	18.5	19.1	20.3	20.0	19.0
Total	72.8	74.1	72.3	73.5	71.8	70.9	74.7	**78.1**	**79.9**	**81.4**	**83.4**	**77.5**	**76.7**	**80.7**

The bold values indicates the result value

consistently shows that KE-LPG achieves higher fluency and clarity compared to the baseline.

4.3.3 Professional Educator Ratings

Table 4 involves seven mathematics educators in assessing lesson plans generated by different models and methods. Evaluation criteria include preliminary analysis, teaching goal clarification, teaching process design, and document specification, ensuring precise analysis of knowledge structure and teaching content correctness. The evaluation assesses tasks analysis accuracy, learner characteristics consideration, and teaching focus identification. Emphasis is on clear objectives aligned with subject characteristics and student cognition, innovation in creating teaching situations, and practical guidance of student participation. Presentation methods, embodiment of new curriculum concepts, and learning methods are also scrutinized, alongside formatting and layout. Evaluation experts are anonymized as Professor A-G, providing valuable insights into lesson plan effectiveness.

4.3.4 Comparison of Different Combining of KG and LLM Methods

To assess the impact of different knowledge graph (KG) integration strategies with large language models (LLMs), we compare our proposed method-GCN-based integration-with three widely used alternatives: GraphSAGE, GAT, and entity linking. The evaluation metrics include the Oread score (textual coherence and fluency) and an educational effectiveness score derived from domain-specific task alignment (see Table 5).

The results demonstrate that our GCN+LLM approach consistently outperforms the baselines, achieving the highest Oread score (72.81%) and educational effectiveness (17.56). These results indicate stronger coherence in generated content as well as superior alignment with educational objectives. In contrast, while GAT and GraphSAGE show moderate improvements over simple entity linking, they fall short in both linguistic quality and instructional value.

Table 5 Comparison of different combinations of KG and LLM

Methods	Oread	Educational-effectiveness
GCN+LLM	**72.81151**	**17.56284**
EntityLink+LLM	70.64482	5.73496
GraphSAGE+LLM	71.02156	5.51818
GAT+LLM	71.65438	5.19783

The bold values indicates the result value

The superior performance of GCN is attributed to its capacity for capturing global and hierarchical relationships within the KG, thereby enabling deeper semantic understanding during content generation. This substantiates the value of graph convolutional structures in integrating structured knowledge with pre-trained language models. Given these findings, we advocate for GCN-based architectures as the preferred strategy in KG-augmented educational language modeling tasks, offering a balanced trade-off between textual fluency and pedagogical accuracy.

4.3.5 Ablation Experiment

In Section 3.1, we introduce a novel methodology that leverages KGs to enhance LLM. Our approach involves using LLMs to extract keywords from the knowledge base and then evaluating relationship weights among these keywords using Graph Laplacian learning techniques. Ablation experiments detailed in Table 6 assess the impact of this method on text readability.

The configuration KE-LPG (Not Enhanced) refers to the ablation setting where the knowledge graph (KG)-based semantic refinement components are removed. Specifically, the model does not incorporate the keyword graph constructed via graph Laplacian learning or the subgraph features derived from the GCN. In this setting, the LLM retains its basic fine-tuning on the lesson plan dataset, but no KG-enhanced knowledge integration is applied. This allows us to isolate the contribution of the KG-based semantic refinement from the underlying LLM capability.

Our experimental results show a significant enhancement in text readability when augmenting LLM capabilities

Table 6 Ablation experimental

Model	Lc	Sc	Oread
ChatGLM2-6B	56.96	1.61	70.72
KE-LPG (ChatGLM)	51.80	2.58	**72.81**
KE-LPG (ChatGLM) (Not Enhanced)	54.36	1.99	71.10
Baichuan	42.39	2.21	73.20
KE-LPG (Baichuan)	39.30	1.86	**76.02**
KE-LPG (Baichuan) (Not Enhanced)	41.02	1.97	74.87

The bold values indicates the result value

Table 7 This table shows the GPU memory usage when using different methods

Methods	GPU memory
ChatGLM2	11.5G
KE-LPG (ChatGLM)	**11.5G**
Baichuan2	27.7G
KE-LPG (Baichuan2)	**27.7G**

The bold values indicates the result value
Comparison of GPU memory helps evaluate the performance differences of different methods regarding resource utilization

through this approach. This underscores the effectiveness of integrating KGs into LLM, improving the quality and coherence of generated text. Consequently, our methodology presents an efficient strategy for utilizing KGs to boost natural language processing tasks.

4.3.6 Performance Testing

Table 7 shows the GPU memory utilization of the various methods we tested. Through a thorough analysis of GPU memory utilization, we can discern the performance distinctions among different methods concerning resource utilization. It is worth noting that ChatGLM2 and the KE-LPG method with the addition of KE-LPG show uniform GPU memory usage of about 11.5G. In contrast, Baichuan2 requires higher GPU memory allocation. After fine-tuning by Lora [80], the peak reached 27.7G, but adding KE-LPG did not significantly increase the memory at all. This comparative analysis not only aids in the judicious selection of models suited for specific application scenarios but also underscores the potential efficiency of KE-LPG in resource-constrained environments. The KE-LPG we designed does not significantly increase the GPU memory but also improves the effect.

5 Discussion

In this paper, we focused on integrating Knowledge Graphs (KG) with Large Language Models (LLMs) to generate personalized and semantically rich lesson plans in mathematics. Our approach, KE-LPG, leverages a GCN-based architecture to model the intricate prerequisite dependencies among mathematical knowledge points. These dependencies reflect logical learning progressions, which are crucial for the design of effective instructional materials. By capturing these relationships, our method supports more coherent and pedagogically aligned lesson plans, ultimately enabling adaptive and targeted teaching strategies.

Compared to traditional template-based or purely LLM-driven generation methods, KE-LPG demonstrates improved semantic coherence and knowledge grounding. This helps mitigate common issues such as factual inaccuracies and vague or generic outputs, thereby reducing the cognitive and manual workload on educators. Our results-measured in terms of readability and expert evaluation-suggest a meaningful improvement in the overall quality of lesson plan generation.

We also acknowledge limitations in our evaluation metrics. The results in Table 3 are reported as raw values without statistical testing. To strengthen reliability, we will include significance analysis (e.g., paired t-tests and bootstrap resampling) in the revised version. Moreover, some metrics, such as educational effectiveness and minedit similarity, were insufficiently defined in the original draft. In the revision, we explicitly introduce these metrics in Section 4.1, including their formal definitions, scoring ranges, and references. Finally, while classical readability metrics such as Kincaid-grade and SMOG were originally developed for English, we justify their use here by adopting the Chinese-adapted implementations provided in the *cntext* library. We also highlight their limitations and note that future work will explore readability measures tailored specifically for Chinese educational content.

The construction and maintenance of high-quality domain-specific KGs can be labor-intensive, and the model's computational complexity may pose scalability challenges in broader applications. Moreover, while KGs are effective in modeling relational information, they do not inherently provide formal semantic validation.

To this end, we recognize the value of ontologies-structured, logic-based knowledge representations that support explicit reasoning and formal validation. Ontology-based AI systems have demonstrated success in contexts where semantic accuracy is critical, such as intelligent tutoring and education diagnostics. These systems define precise conceptual hierarchies and rules, enabling stronger guarantees of content validity.

Compared to ontologies, KGs offer more flexibility and scalability, particularly in real-world educational settings where data is often semi-structured or evolving. KGs also integrate more easily with LLMs using existing graph neural network architectures. Nonetheless, we acknowledge that the

lack of formal logical consistency mechanisms in KGs may limit the semantic precision of LLM outputs.

Recent studies have explored hybrid approaches that integrate ontologies with LLMs, combining the interpretability and reasoning power of ontologies with the adaptability and language understanding of modern language models. Inspired by this, future work will explore such hybrid frameworks, aiming to retain the flexibility of KG-based modeling while benefiting from the formal guarantees of ontological reasoning. This direction could address current limitations in validation and semantic accuracy.

Furthermore, although our current work focuses on mathematics, our approach has the potential to be applied across other academic disciplines. Expanding to a multidisciplinary context will require the construction of new KGs and the adaptation of evaluation strategies to fit domain-specific pedagogical requirements. Improving our evaluation metrics and incorporating more diverse datasets will further enhance the robustness of the model and its applicability in real-world educational environments.

While our experiments show 2% improvements in text readability and 5–6% in expert ratings, we acknowledge that this level of accuracy may still fall short of expectations for AI systems in critical domains. Education, despite being different from sectors like healthcare or nuclear safety, plays a foundational role in societal development. Thus, the precision of AI-generated educational content should be held to a similarly high standard. Our work offers an early step toward achieving this goal and lays the groundwork for more reliable, semantically aware educational AI systems.

6 Conclusion and Future Work

In this study, we proposed KE-LPG (Knowledge-Enhanced Lesson Plan Generation), a novel approach that effectively integrates Knowledge Graphs (KG) with Large Language Models (LLM) to generate high-quality, semantically grounded, and personalized instructional plans. By leveraging the structured semantic representation of KGs and the generative capabilities of LLMs, KE-LPG offers a powerful framework for educational content generation that balances factual accuracy, domain relevance, and contextual fluency. Our method was empirically validated using a comprehensive three-year mathematics curriculum dataset, further enriched with internship-based instructional records. This fusion of structured and real-world educational data enables KE-LPG to support pedagogically sound lesson planning that aligns with cognitive development frameworks such as Bloom's Taxonomy. The resulting lesson plans demonstrate improved coherence, adaptability, and instructional relevance, marking a significant advancement in the application of AI in education.

While our findings are promising, several limitations remain. The model currently depends on the availability and quality of domain-specific knowledge graphs, and the computational demands of integrating GCN with LLM present scalability challenges. Additionally, although our evaluation metrics capture core aspects of instructional effectiveness, further refinement—particularly incorporating student feedback and longitudinal outcomes—would strengthen the assessment of pedagogical impact.

Looking forward, future research will explore the incorporation of multimodal data (e.g., video lectures, assessments), expand the model to diverse subject domains, and enhance interpretability and ethical alignment of the generated plans. Moreover, we aim to develop more robust evaluation frameworks that better reflect real-world classroom dynamics and instructional diversity.

Acknowledgements We acknowledge the support of the National Natural Science Foundation of China under Grant (No. 62577050), and the Natural Science Foundation of Zhejiang Province under Grant (No. LY23F020010)

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Schmid, M.; Brianza, E.; Petko, D.: Self-reported technological pedagogical content knowledge (tpack) of pre-service teachers in relation to digital technology use in lesson plans. Comput. Hum. Behav. **115**, 106586 (2021)
2. Mok, S.Y.; Staub, F.C.: Does coaching, mentoring, and supervision matter for pre-service teachers' planning skills and clarity of instruction? a meta-analysis of (quasi-) experimental studies. Teach. Teach. Educ. **107**, 103484 (2021)
3. Moundridou, M.; Matzakos, N.; Doukakis, S.: Generative ai tools as educators' assistants: designing and implementing inquiry-based lesson plans. Comput. Educ. Artif. Intell. **7**, 100277 (2024)
4. Ormell, C.P.: Bloom's taxonomy and the objectives of education. Educ. Res. **17**(1), 3–18 (1974)
5. Anderson, L.W.; Krathwohl, D.R.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Complete Addison Wesley Longman Inc, Saddle River (2001)
6. Gunawardena, M.; Bishop, P.; Aviruppola, K.: Personalized learning: the simple, the complicated, the complex and the chaotic. Teach. Teach. Educ. **139**, 104429 (2024)
7. Wang, Q.; Zhu, J.; Pan, C.; Shi, J.; Meng, C.; Guo, H.: Dual trustworthy mechanism for illness classification with multi-modality data. In: 2023 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 356–362. IEEE (2023)
8. Bakker, A.B.; Mostert, K.: Study demands-resources theory: understanding student well-being in higher education. Educ. Psychol. Rev. **36**(3), 92 (2024)
9. Huang, C.; Tu, Y.; Wang, Q.; Li, M.; He, T.; Zhang, D.: How does social support detected automatically in discussion forums relate

to online learning burnout? The moderating role of students'self-regulated learning. Comput. Educ. **227**, 105213 (2025)
10. Hassen, S.B.; Neji, M.; Hussain, Z.; Hussain, A.; Alimi, A.M.; Frikha, M.: Deep learning methods for early detection of alzheimer's disease using structural mr images: a survey. Neurocomputing **576**, 127325 (2024)
11. Albaseer, A.; Abdallah, M.; Al-Fuqaha, A.: Exploiting the divergence between output of ml models to detect adversarial attacks in streaming IoT applications. In: ICC 2023-IEEE International Conference on Communications, pp. 3090–3095. IEEE (2023)
12. Wu, T.; He, S.; Liu, J.; Sun, S.; Liu, K.; Han, Q.-L.; Tang, Y.: A brief overview of chatgpt: the history, status quo and potential future development. IEEE/CAA J. Autom. Sinica **10**(5), 1122–1136 (2023)
13. Elgendy, I.; Muthanna, A.; Hammoudeh, M.; Shaiba, H.A.; Unal, D.; Khayyat, M.: Security-aware data offloading and resource allocation for mec systems: a deep reinforcement learning. Authorea Preprints (2023)
14. Zhu, J.; Guo, H.; Shi, W.; Chen, Z.; De Meo, P.: Radio: Real-time hallucination detection with contextual index optimized query formulation for dynamic retrieval augmented generation. Proc. the AAAI Conf. Artif. Intelli. **39**, 26129–26137 (2025)
15. Guo, H.; Zhu, J.; Di, S.; Shi, W.; Chen, Z.; Xu, J.: Dior: Adaptive cognitive detection and contextual retrieval optimization for dynamic retrieval-augmented generation. arXiv preprint arXiv:2504.10198 (2025)
16. Chi, H.; Li, H.; Yang, W.; Liu, F.; Lan, L.; Ren, X.; Liu, T.; Han, B.: Unveiling causal reasoning in large language models: Reality or mirage? Adv. Neural. Inf. Process. Syst. **37**, 96640–96670 (2024)
17. Su, W.; Tang, Y.; Ai, Q.; Wu, Z.; Liu, Y.: Dragin: dynamic retrieval augmented generation based on the information needs of large language models. arXiv preprint arXiv:2403.10081 (2024)
18. Schramm, S.; Wehner, C.; Schmid, U.: Comprehensible artificial intelligence on knowledge graphs: a survey. J. Web Semant. **79**, 100806 (2023)
19. Yang, X.; Chen, Z.; Guo, H.; Shestakevych, T.: Adaptive exploration: elevating educational impact of unsupervised knowledge graph question answering. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, pp. 239–248. Springer (2024)
20. Zhu, J.; Ma, X.; Huang, C.: Stable knowledge tracing using causal inference. IEEE Trans. Learn. Technol. **17**, 124–134 (2023)
21. Wang, X.; Chen, Z.; Wang, H.; Hou U. L.; Li, Z.; Guo, W.: Large language model enhanced knowledge representation learning: a survey. Data Sci. Eng. 1–24 (2025)
22. Li, R.; Di, S.; Chen, L.; Zhou, X.: Simdiff: Simple denoising probabilistic latent diffusion model for data augmentation on multi-modal knowledge graph. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1631–1642 (2024)
23. Chen, Z.; Wang, Y.; Zhao, B.; Cheng, J.; Zhao, X.; Duan, Z.: Knowledge graph completion: a review. IEEE Access **8**, 192435–192456 (2020)
24. Zhong, L.; Wu, J.; Li, Q.; Peng, H.; Wu, X.: A comprehensive survey on automatic knowledge graph construction. ACM Comput. Surv. **56**(4), 1–62 (2023)
25. Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; Liu, X.; Sun, F.; He, K.: A survey of knowledge graph reasoning on graph types: static, dynamic, and multi-modal. IEEE Trans. Pattern Anal. Mach. Intell. (2024)
26. Wan, Y.; Liu, Y.; Chen, Z.; Chen, C.; Li, X.; Hu, F.; Packianather, M.: Making knowledge graphs work for smart manufacturing: research topics, applications and prospects. J. Manuf. Syst. **76**, 103–132 (2024)
27. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
28. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
29. Ma, T.; Pan, Q.; Rong, H.; Qian, Y.; Tian, Y.; Al-Nabhan, N.: T-bertsum: topic-aware text summarization based on Bert. IEEE Trans. Comput. Soc. Syst. **9**(3), 879–890 (2021)
30. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
31. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
32. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
33. Cifarelli, C.P.; Sheehan, J.P.: Large language model artificial intelligence: the current state and future of chatgpt in neuro-oncology publishing. J. Neurooncol. **163**(2), 473–474 (2023)
34. OpenAI, R.: Gpt-4 technical report. arxiv 2303.08774. View in Article 2, 3 (2023)
35. Vemprala, S.; Bonatti, R.; Bucker, A.; Kapoor, A.: Chatgpt for robotics: design principles and model abilities. Microsoft Auton. Syst. Robot. Res **2**, 20 (2023)
36. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
37. Ye, H.; Liu, T.; Zhang, A.; Hua, W.; Jia, W.: Cognitive mirage: a review of hallucinations in large language models. CoRR arXiv:2309.06794 (2023)
38. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12), 1–38 (2023)
39. Tiddi, I.; Schlobach, S.: Knowledge graphs as tools for explainable machine learning: a survey. Artif. Intell. **302**, 103627 (2022)
40. Li, Y.; Wu, Y.; Zhang, L.; Chen, J.: The impact of network structure on knowledge adoption: a network text analysis on knowledge-sharing platforms. IEEE Trans. Comput. Soc. Syst. (2023)
41. Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; Stone, P.: Llm+ p: empowering large language models with optimal planning proficiency. arXiv preprint arXiv:2304.11477 (2023)
42. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X.: Unifying large language models and knowledge graphs: a roadmap. arXiv preprint arXiv:2306.08302 (2023)
43. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q.: Ernie: enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129 (2019)
44. Tian, H.; Gao, C.; Xiao, X.; Liu, H.; He, B.; Wu, H.; Wang, H.; Wu, F.: Skep: sentiment knowledge enhanced pre-training for sentiment analysis. arXiv preprint arXiv:2005.05635 (2020)
45. Zhang, C.-Y.; Fang, W.-P.; Cai, H.-C.; Chen, C.P.; Lin, Y.-N.: Sparse graph transformer with contrastive learning. IEEE Trans. Comput. Soc. Syst. **11**(1), 892–904 (2022)
46. Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J.: Qa-gnn: Reasoning with language models and knowledge graphs for question answering. arXiv preprint arXiv:2104.06378 (2021)
47. Liu, X.; Meng, S.; Li, Q.; Liu, Q.; He, Q.; Ramesh, D.; Qi, L.: Fdgnn: feature-aware disentangled graph neural network for recommendation. IEEE Trans. Comput. Soc. Syst. **11**(1), 1372–1383 (2023)

48. Kang, M.; Baek, J.; Hwang, S.J.: Kala: knowledge-augmented language model adaptation. arXiv preprint arXiv:2204.10555 (2022)
49. Zhang, T.; Wang, C.; Hu, N.; Qiu, M.; Tang, C.; He, X.; Huang, J.: Dkplm: decomposable knowledge-enhanced pre-trained language model for natural language understanding. Proc. AAAI Conf. Artif. Intell. **36**, 11703–11711 (2022)
50. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P.: K-bert: enabling language representation with knowledge graph. Proc. AAAI Conf. Artif. Intell. **34**, 2901–2908 (2020)
51. Sun, T.; Shao, Y.; Qiu, X.; Guo, Q.; Hu, Y.; Huang, X.; Zhang, Z.: Colake: contextualized language and knowledge embedding. arXiv preprint arXiv:2010.00309 (2020)
52. Rosset, C.; Xiong, C.; Phan, M.; Song, X.; Bennett, P.; Tiwary, S.: Knowledge-aware language model pretraining. arXiv preprint arXiv:2007.00655 (2020)
53. Backfisch, I.; Lachner, A.; Hische, C.; Loose, F.; Scheiter, K.: Professional knowledge or motivation? investigating the role of teachers' expertise on the quality of technology-enhanced lesson plans. Learn. Instr. **66**, 101300 (2020)
54. Touretzky, D.; Gardner-McCune, C.; Martin, F.; Seehorn, D.: Envisioning ai for k-12: What should every child know about AI? Proc. AAAI Conf. Artif. Intell. **33**, 9795–9799 (2019)
55. Chen, L.; Chen, P.; Lin, Z.: Artificial intelligence in education: a review. IEEE Access **8**, 75264–75278 (2020)
56. Paiva, J.C.; Leal, J.P.; Figueira, Á.: Automated assessment in computer science education: a state-of-the-art review. ACM Trans. Comput. Educ. (TOCE) **22**(3), 1–40 (2022)
57. Chen, X.; Zou, D.; Xie, H.; Wang, F.L.: Past, present, and future of smart learning: a topic-based bibliometric analysis. Int. J. Educ. Technol. High. Educ. **18**(1), 2 (2021)
58. Lin, C.-C.; Huang, A.Y.; Lu, O.H.: Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. Smart Learn Environ **10**(1), 41 (2023)
59. Glaser, R.: Cognitive psychology and instructional design. In: Cognition and Instruction, pp. 303–315. Psychology Press, Hove (2014)
60. Wang, Z.; Manning, K.; Mallick, D.B.; Baraniuk, R.G.: Towards blooms taxonomy classification without labels. In: Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part I 22, pp. 433–445. Springer (2021)
61. Coşgun Ögeyik, M.: Using bloom's digital taxonomy as a framework to evaluate webcast learning experience in the context of covid-19 pandemic. Educ. Inf. Technol. **27**(8), 11219–11235 (2022)
62. Tang, X.; Chen, Y.; Li, X.; Liu, J.; Ying, Z.: A reinforcement learning approach to personalized learning recommendation systems. Br. J. Math. Stat. Psychol. **72**(1), 108–135 (2019)
63. Horner, R.H.; Ward, C.S.; Fixsen, D.L.; Sugai, G.; McIntosh, K.; Putnam, R.; Little, H.D.: Resource leveraging to achieve large-scale implementation of effective educational practices. J. Posit. Behav. Interv. **21**(2), 67–76 (2019)
64. Viswan, V.; Shaffi, N.; Mahmud, M.; Subramanian, K.; Hajamohideen, F.: Explainable artificial intelligence in Alzheimer's disease classification: a systematic review. Cogn. Comput. **16**(1), 1–44 (2024)
65. Nordin, N.M.; Klobas, J.: Wikis as collaborative learning tools for knowledge sharing: shifting the education landscape. In: Global Learn, pp. 331–340. Association for the Advancement of Computing in Education (AACE) (2010)
66. Abi-Rafeh, J.; Hanna, S.; Bassiri-Tehrani, B.; Kazan, R.; Nahai, F.: Complications following facelift and neck lift: implementation and assessment of large language model and artificial intelligence (chatgpt) performance across 16 simulated patient presentations. Aesthetic Plast. Surg. **47**(6), 2407–2414 (2023)
67. Yu, Q.; Zhang, J.; Zhang, H.; Wang, Y.; Lin, Z.; Xu, N.; Bai, Y.; Yuille, A.: Mask guided matting via progressive refinement network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1154–1163 (2021)
68. Xiong, H.; Bian, J.; Li, Y.; Li, X.; Du, M.; Wang, S.; Yin, D.; Helal, S.: When search engine services meet large language models: visions and challenges. IEEE Trans. Serv. Comput. **17**, 4558–4577 (2024)
69. Cao, B.; Deng, H.; Hao, Y.; Luo, X.: Multi-view information fusion based on federated multi-objective neural architecture search for MRI semantic segmentation. Inf. Fusion **123**, 103301 (2025)
70. Yang, E.; Zhou, W.; Qian, X.; Lei, J.; Yu, L.: Drnet: dual-stage refinement network with boundary inference for rgb-d semantic segmentation of indoor scenes. Eng. Appl. Artif. Intell. **125**, 106729 (2023)
71. Saha, S.; Obukhov, A.; Paudel, D.P.; Kanakis, M.; Chen, Y.; Georgoulis, S.; Van Gool, L.: Learning to relate depth and semantics for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8197–8207 (2021)
72. Kim, B.; Han, S.; Kim, J.: Discriminative region suppression for weakly-supervised semantic segmentation. Proc. AAAI Conf. Artif. Intell. **35**, 1754–1761 (2021)
73. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.-W.: Unified language model pre-training for natural language understanding and generation. Adv. Neural Inf. Process. Syst. **32** (2019)
74. Andrus, B.R.; Nasiri, Y.; Cui, S.; Cullen, B.; Fulda, N.: Enhanced story comprehension for large language models through dynamic document-based knowledge graphs. Proc. AAAI Conf. Artif. Intell. **36**, 10436–10444 (2022)
75. Bansal, G.; Chamola, V.; Hussain, A.; Guizani, M.; Niyato, D.: Transforming conversations with ai-a comprehensive study of chatgpt. Cogn. Comput. 1–24 (2024)
76. Liu, Q.; Dong, Y.; Li, X.: Multi-stage context refinement network for semantic segmentation. Neurocomputing **535**, 53–63 (2023)
77. Brüggemann, D.; Sakaridis, C.; Truong, P.; Van Gool, L.: Refign: align and refine for adaptation of semantic segmentation to adverse conditions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 3174–3184 (2023)
78. Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al.: Baichuan 2: open large-scale language models. arXiv preprint arXiv:2309.10305 (2023)
79. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J.: Glm: General language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360 (2021)
80. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.: Lora: low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)