PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

ChemKGL: Bridging Knowledge Graphs and Large Language Models for Chemical Multi-Step Reaction Pathway Inference

Fan Yang[1] , **Feiyang Xu**[1,2] , **Kun Zhang**[1,3] , **Huadong Liang**[2] , **Pengyang Shao**[4] , **Xin Li**[5,6,*] and **Le Wu**[1,*]
[1]Hefei University of Technology, China
[2]Artiffcial Intelligence Research Institute, iFLYTEK Co., Ltd., China
[3]Intelligent Interconnected Systems Laboratory of Anhui Province (HFUT), China
[4]National University of Singapore, Singapore
[5]University of Science and Technology of China, China.
[6]State Key Laboratory of Cognitive Intelligence, China.
leexin@ustc.edu.cn, lewu@hfut.edu.cn

IJCAI-ECAI 2026 PREPRINT

Abstract

Large language models have shown promising potential in chemistry, with prior work exploring molecular recognition, classification, and property prediction. Despite the achieved progress, LLMs are still far from satisfactory when dealing with complex chemical multi-step reaction pathway inference task due to the *lack of domain knowledge* and *limited ability to maintain consistent multi-step reasoning*. To address these challenges, we propose ChemKGL, a novel multi-step reasoning framework enhanced by knowledge graph retrieval. We reformulate the task as a knowledge graph retrieval problem to better control LLM generation, and leverage graph technologies to integrate textual descriptions, reaction conditions, and materials. Based on this reformulation, we design a novel multi-retrieval strategy to improve the performance of reaction pathway inference, which includes forward retrieval, reverse retrieval, and historical retrieval. An entropy-based pruning mechanism is further introduced to alleviate cold-start issues caused by retrieval failure. In addition, we construct a comprehensive chemical knowledge graph and corresponding dataset to support model training and analysis. Finally, we employ multiple evaluation methods to assess our proposed ChemKGL, and the experimental results demonstrate its superiority. Our code and dataset can be obtained at https://github.com/Double-Sail/ChemKGL.

1 Introduction

The rapid development of large language models (LLMs) has continuously advanced research in the field of chemistry, especially in tasks requiring complex reasoning. Among these, chemical multi-step reaction pathway inference task (CMSR-PIT) has become a research focus due to its strong reliance on step-by-step logical inference. Unlike tasks such as molecular recognition [Bhattacharya *et al.*, 2024], molecular classification [Edwards *et al.*, 2022], or property prediction [Yang *et al.*, 2019], this task requires the model to infer multiple intermediate steps and accurately understand the logical relationships between them. This places higher demands on the depth and coherence of reasoning. To improve the performance of LLMs in such tasks, current research mainly focuses on two directions. The first is the introduction of the Chain-of-Thought (CoT) mechanism, which guides the model to reason step by step [Wei *et al.*, 2022]. This approach aligns well with the stepwise nature of multi-step chemical reactions. The second is the integration of structured knowledge graphs, which support reasoning process with clear entity relationships and strong interpretability [Bai *et al.*, 2025; Ma *et al.*, 2025].

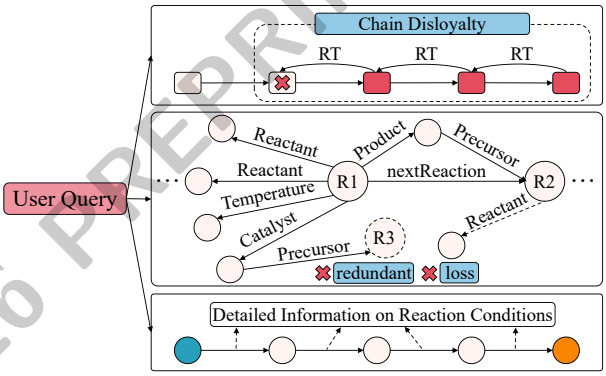

Figure 1: Deficiencies of traditional CoT and knowledge graph on multi-step chemical reaction pathway reasoning. RT stands for Reflective Thinking, and R_n stands for the Reaction n.

Despite the achieved progress, some recent studies argue that CoT reasoning might act as pseudo-logic, failing to substantially improve true reasoning capabilities [Tanneru *et al.*, 2024; Jin *et al.*, 2024]. As illustrated in the "Chain Disloyalty" module at the top of Figure 1, in longer reasoning chains, the model often reinforces early errors through repeated "reflective thinking". This leads to a phenomenon known as "chain disloyalty", where the model generates

logically coherent but factually incorrect reasoning paths. Such paths can obscure hallucination issues and reduce interpretability [Lu *et al.*, 2025]. On the other hand, current research applying knowledge graphs to chemical reasoning tasks typically represents reaction conditions—such as catalysts, solvents, and temperatures—as nodes, connected using simple relational properties (e.g., "hasReactant" "usesCatalyst" and "isPrecursorOf" in Figure 1). This graph structure presents two main problems. First, the oversimplified relationship design makes retrieval and pruning inefficient and imprecise, leading to information loss (e.g., "Reaction 2" is missing a reactant due to retrieval failure) or redundancy (e.g., "Reaction 3" is incorrectly included due to pruning failure). Second, the resulting subgraphs are often multi-branched and lack clear temporal order and linear pathways (e.g., "Reaction 1" and "Reaction 2" are only connected via "nextReaction" in Figure 1), which does not align with the chain-like evolution of multi-step chemical reactions. These issues lead to a central challenge: "**How can we efficiently obtain an informative and temporally ordered reaction pathway to support interpretable multi-step reasoning in LLMs?**"

To address the above issues, we first re-formulate CMSRPIT as the knowledge graph retrieval, which can effectively control the LLM generation. Meanwhile, we collected a series of real-world multi-step chemical reactions through crowdsourcing, and construct a knowledge graph with question-answer pairs for model tuning and evaluation. Then, we propose a novel ChemKGL, a knowledge graph retrieval-augmented framework designed to enhance the reasoning ability of LLMs in the CMSRPIT. Specifically, ChemKGL includes three main components. Firstly, ChemKGL uses an advanced LLM to identify chemical entities in the user query, retaining only reactants and products. Second, ChemKGL develops a robust retrieval strategy to perform subgraph-level retrieval in the knowledge graph based on the extracted entities. This strategy includes triple retrieval components: reverse retrieval, forward retrieval, and historical retrieval, aiming to ensure complete coverage of the reaction context. Meanwhile, this strategy includes a fault-tolerant mechanism to handle extreme cases such as missing entities or broken paths. Third, the retrieval subgraph is encoded with two modalities: textual content and structural encoding. ChemKGL fuses the encoded embeddings and sends to an LLM for complex reaction pathways generation. Finally, we conduct evaluations using a state-of-the-art LLM evaluation team and an open-source benchmark, and the evaluation results of the LLM are reviewed by experts. Through these two methods, ensure the rationality and reliability of the evaluation. Extensive evaluation on real-world data demonstrates the superiority of our proposed ChemKGL.

2 Related Work

2.1 Chemical Multi-step Reaction Pathway Inference Task

In recent years, advances in large language models (LLMs) have accelerated chemistry-related research [Zhong *et al.*, 2024; M. Bran *et al.*, 2024]. However, most existing studies focus on single-hop knowledge retrieval and lack deep exploration of complex reasoning. In contrast, multi-step chemical reasoning tasks—such as retrosynthesis planning—are essential for material discovery, chemical process optimization, and interdisciplinary research [Chen *et al.*, 2020]. Although several works have applied LLMs to retrosynthesis [Yu *et al.*, 2024; Kim *et al.*, 2021], they typically construct multi-step routes by concatenating single-step reactions, which raises concerns about the feasibility of the resulting pathways.

Moreover, the diversity and uncertainty of retrosynthesis objectives require models to expend substantial computational effort to decide when to terminate decomposition, distracting them from core reasoning. By comparison, our dataset is based on real multi-step reaction processes. Each task explicitly specifies the starting materials and target product, enabling the model to focus on intermediate reasoning steps. This formulation also simplifies evaluation and better reflects practical scenarios, such as synthesizing a target compound from available laboratory reagents.

2.2 Knowledge Graph Retrieval-Augmented Generation

Model interpretability has long been a central topic in deep learning research. Darren et al. [Edge *et al.*, 2025] proposed GraphRAG to address the limitations of traditional RAG methods when target knowledge is distributed across multiple chunks, improving interpretability and demonstrating the effectiveness of knowledge graph retrieval for enhancing LLM reasoning. In recent years, several related approaches have emerged. Zhou et al. [Zhou *et al.*, 2025] introduced RefKG, which improves reasoning through reflective interaction with knowledge graphs. Mavromatis et al. [Mavromatis and Karypis, 2025] proposed GNN-RAG, integrating graph neural networks with retrieval-augmented generation for knowledge graph question answering. Song et al. [Song *et al.*, 2025] presented REKG-MCTS, achieving strong performance on complex knowledge graph reasoning tasks.

However, these methods primarily focus on global or direct relational reasoning, limiting their ability to satisfy complex conditional requirements in the chemical domain. This limitation is particularly evident in macromolecular reactions, where reactants and products strongly depend on rich reaction conditions. Knowledge graphs with overly simplistic relations often fail to capture such information. Our method effectively addresses this gap.

3 Technical Details of ChemKGL

As shown in Figure 2, our proposed ChemKGL consists of three main components: 1) *entity extraction and recognition*: extracting entity information from user questions, identifying reactants and products, and standardizing their SMILES representations; 2) *subgraph retrieval*: developing a retrieval framework that produces a chain-structured reaction subgraph; 3) *encoding fusion and inference*: fusing the question and subgraph encodings and sending to LLMs for generation. Next, we will introduce each component in detail.

3.1 Entity Extraction and Recognition

User queries often contain excessive entity information, such as catalysts, solvents, temperature, and pressure. When or-

IJCAI-ECAI 2026 PREPRINT

PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

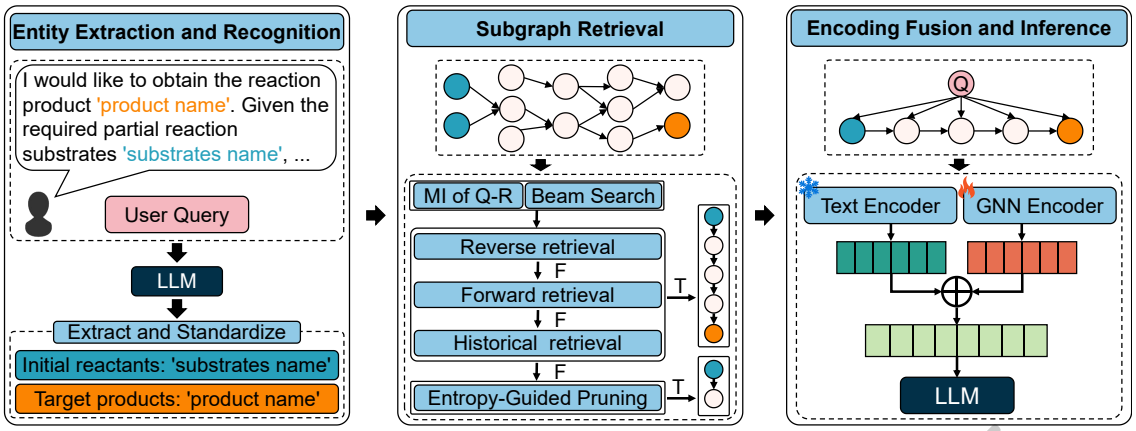

Figure 2: The overall framework of ChemKGL. In the first part, "Extract and Standardize" denotes the standardization of SMILES representations for the extracted compounds. In the second part, "MI of Q-R" represents mutual information of user questions and node relationships. In the third part, Text Encoder is frozen and GNN Encoder is trainable.

ganizing these contents in knowledge graph, too many node types will have a negative impact on LLM reasoning. Thus, we only focus on the initial reactant and target product entities, improving the query understanding and utilization.

Although there are many open-source named entity recognition (NER) models available [Yadav and Bethard, 2018; Huang *et al.*, 2023], they do not directly meet our specific needs. In particular, these models lack the ability to distinguish between reactants and products among the extracted entities. Addressing this usually requires building supervised datasets and fine-tuning models, which is both costly and time-consuming. Fortunately, recent advances show that LLMs have achieved impressive performance on NER tasks, even under low-resource and few-shot settings [Wang *et al.*, 2025]. Therefore, we adopt a prompt-tuning approach, using carefully designed prompt templates to guide the LLM in accurately extracting and classifying entities. Details of the prompt templates are provided in the Appendix[1] A.1. In addition, we use the RDKit [Landrum, 2023] interface to standardize the extracted SMILES representations, enabling accurate matching with nodes in the knowledge graph.

3.2 Subgraph Retrieval

Based on mutual information retrieval

Existing studies show shared latent concepts between prompt and query are critical for activating in-context learning [Xie *et al.*, 2022], yet such concepts are not directly observable. To tackle this, inspired by the knowledge editing work of Shi et al. [Shi *et al.*, 2024], we propose maximizing mutual information $I(Q; G_S)$ between user query Q and retrieved subgraph G_S. This indirectly aligns latent concepts and boosts reasoning performance. The optimization objective is formulated in Eq. 1, where H denotes Shannon entropy.

$$\max_{G_S} I(Q; G_S) = H(Q) - H(Q \mid G = G_S). \quad (1)$$

[1]The appendix is available at https://github.com/Double-Sail/ChemKGL.

Focusing on the initial reactant and target product entities, the retrieved subgraph G_S forms a linear structure containing only reactants, products, and reaction conditions. This chain-like structure simplifies the computation of $I(Q; G_S)$, which can be performed iteratively. At each step, only the conditional probability p of the relation text in the next triple needs to be estimated based on current information. Given the strong next-word prediction capability of pretrained LLMs [Vaswani *et al.*, 2017], we use them to estimate this conditional probability. Combined with the rich relational information in our knowledge graph, this relation-based subgraph retrieval method is well suited to our task. Detailed derivations are provided in Appendix A.4.

Finally, since this iterative process follows a greedy strategy and may suffer from local optima, we adopt a beam search approach [Medress *et al.*, 1977] that maintains multiple candidate paths and selects the one with the highest mutual information as the final result.

Hybrid Reasoning Retrieval

To address the issue that a single retrieval strategy may not ensure sufficient accuracy, we propose a hybrid retrieval approach. This strategy is inspired by how humans solve reasoning tasks with known starting and ending points—using forward reasoning, reverse reasoning, and leveraging historical references. Based on experimental accuracy, we assign priorities to each strategy and apply them in descending order to reduce unnecessary computation and resource consumption while improving retrieval performance and efficiency.

We start with reverse retrieval as the initial strategy. It begins from the target product node and performs step-by-step reasoning until the starting reactant node is found. Previous studies have shown that reverse retrieval can effectively enhance the reasoning ability of LLMs [Chen *et al.*, 2025; Liu *et al.*, 2025b]. If the starting node is not found within the maximum number of steps, the process moves to forward retrieval, which starts from the known reactant and attempts to reach the target product. If forward retrieval also fails, we turn to the third stage—history-assisted forward retrieval (i.e.,

IJCAI-ECAI 2026 PREPRINT

PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

historical retrieval). In this stage, each triple in the graph includes "historical product" information, indicating which products were generated by that reaction. This additional context helps strengthen the mutual information between intermediate nodes and the target product, improving retrieval accuracy in the middle steps of reasoning.

We introduce historical information because we observed that forward retrieval usually performs well in the first step—even when a node has thousands of child nodes, mutual information can still guide the model to the correct path. However, failures often occur during intermediate steps, where direct semantic links to the target are weaker. This is especially true when using general-purpose open-source models, which struggle with mutual information estimation in such cases. Historical product data helps fill this gap by enhancing the relevance of intermediate reasoning steps.

It is important to note that reverse retrieval does not use historical information, as all its steps inherently point toward the target product, making such information redundant. Also, although history-assisted retrieval is an extension of forward retrieval, we still retain the regular forward stage. This is because a single reaction may lead to multiple products, and adding all historical data at once may greatly increase token usage and computational cost. Therefore, historical retrieval serves as a backup strategy when the first two fail. In summary, our hybrid retrieval process follows this order: reverse retrieval $\rightarrow$ forward retrieval $\rightarrow$ historical retrieval, ensuring both reasoning diversity and efficient resource use.

Entropy-Based Pruning

To address the limitations of single-mode retrieval, which cannot always ensure accuracy, we propose a hybrid retrieval approach. However, in extreme cases, all retrieval strategies may fail. In some instances, the target product required by the user may not exist in the knowledge graph, inevitably leading to retrieval failure and preventing the return of the required subgraph. This leads to a cold start problem for the model.

To mitigate the impact of cold start problem, when historical retrieval fails, we still provide the model with a reasoning path subgraph $G_S = \{a_1, a_2, \ldots, a_n\}$, where $a_i = \{h_i, r_i, t_i\}$, and h represents the head node, r represents the relationship, and t represents the tail node. Since the tail entity t_n in the final triple is not the target product, feeding the entire reasoning path into the model may introduce noise. However, we observe that the head entity h_1 in the first triple matches the known reactant in the query. This suggests that the early part of the reasoning chain may still be informative. Therefore, we introduce a pruning strategy based on uncertainty estimation to extract the useful portion and remove redundant information. Specifically, we use Shannon entropy to measure the uncertainty of the output given different inputs, as shown in Eq. 2. Where X represents the input and Y represents its corresponding output, the model's uncertainty is quantified by the entropy value H as follows:

$$H(Y \mid X = x) = -\sum_{y} p(y \mid x) \log_2 p(y \mid x), \qquad (2)$$

where y represents each possible answer generated by the LLM, we construct a series of inputs $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i = \{Q, a_1, a_2, \ldots, a_i\}$ represents an incremental reasoning path. For input x_i, we compute the entropy H_i, obtaining an entropy set $\hat{H} = \{H_1, H_2, \ldots, H_n\}$. A lower entropy indicates that the model is more confident in its response. Finally, we prune the reasoning path at the position where entropy is minimized. We retain all steps before this point, giving us $\hat{H}_{min} = \{H_1, H_2, \ldots, H_{min}\}$, and the corresponding subgraph $G_{Smin} = \{a_1, a_2, \ldots, a_{min}\}$. This subgraph is considered to provide the most informative and least noisy input to the model.

3.3 Encoding Fusion and LLM Inference

Knowledge graphs are highly structured data carriers. The semantic and topological information embedded in nodes and edges can be weakened or even lost during simple text transformation. Therefore, relying solely on text representations is not enough to fully leverage the structural knowledge in graphs. Previous research has shown that combining LLMs with graph neural networks (GNNs) can effectively integrate both textual semantics and graph structures, leading to improved reasoning performance [Yasunaga *et al.*, 2021].

Therefore, we propose a fusion encoding strategy. First, we take retrieved subgraph $G_S = \{a_1, a_2, \ldots, a_n\}$, where each edge represents a relation r_i. All relations are concatenated to form a complete reasoning chain as text $T = [r_1, r_2, \ldots, r_n]$. Since each relation r_i already contains rich semantic information, the nodes themselves are not included in the text encoding. We input the text sequence T into an embedding model, obtaining the textual representation.

$$E_{text} = Emb(T) = \{e_1, e_2, \ldots, e_n\}. \qquad (3)$$

Next, as shown in the third part of Figure 2, we represent the question as a special node Q and construct a working subgraph G_w by combining it with the retrieved subgraph G_S. We then apply a Graph Attention Network (GAT) [Velickovic *et al.*, 2017] on G_w to update the node representations. GAT captures the semantic and structural dependencies between each node and its neighbors by computing attention weights, resulting in the graph-based encoding:

$$E_{gnn} = GAT(G_w). \qquad (4)$$

Finally, we concatenate the two encodings to form the final representation $E_{final} = [E_{text}; E_{gnn}]$, which serves as a hard prompt for the LLM to generate the reasoning output.

4 Experiment

4.1 Experimental Setup

Dataset Construction

We collected detailed information on a series of multi-step chemical reactions through crowdsourcing and organized the data in a structured format. The data were sourced from published multi-step reaction studies in leading journals such as Nature Chemistry and Chemical Science, as well as from publicly available patents involving multi-step chemical reactions. Unlike commonly used USPTO-based [Lowe, 2012] datasets in the field, our dataset consists of genuine multi-step chemical reactions rather than sequences artificially assembled from individual single-step reactions. Based on this,

IJCAI-ECAI 2026 PREPRINT

Model Name	Parameters	GPT4-o1	Deepseek-R1	Gemini 2.5Pro	Average
LLaMA3.1	8B	45.40	26.74	10.23	27.46
Qwen2.5	7B	59.41	30.68	13.28	34.46
Qwen2.5	72B	83.81	42.44	21.81	49.35
Qwen3	8B	71.12	34.48	16.95	40.85
Deepseek-V3	671B	90.97	49.52	38.71	59.73
LLaMA3.1-ChemKGL	8B	84.51	58.51	69.19	**70.74**
Qwen2.5-ChemKGL	7B	83.04	54.99	64.02	<u>67.35</u>
LLaMA3.1-REKG-MCTS	8B	76.94	55.25	50.24	60.81
Qwen2.5-REKG-MCTS	7B	80.00	55.90	51.59	62.50

Table 1: Experimental results for 120-Sample Test Set. The best results are in **bold** and the second-best are <u>underlined</u>.

we selected reaction cases with 2 to 5 steps to construct a training set of 20,000 question-answer pairs. To improve the model's ability to understand different expressions, we rewrote each question in four diverse ways. Specific examples can be found in the Appendix A.2. And we invited multiple chemical experts to select 30 more challenging cases for each reaction step (a total of 120 cases) to construct a test set.

In addition, to further enhance the credibility and persuasiveness of the experimental results, we also conducted supplementary experiments on the oMeBench [Xu *et al.*, 2025]. It comprises over 10,000 annotated mechanistic steps with intermediates, type labels, and difficulty ratings. This benchmark dataset characterizes another type of multi-hop reactive reasoning task and was recently released, effectively reducing the possibility of the model being exposed to related data during the training stage.

Evaluation method

Research by Zheng et al. has proven that, in the absence of a "golden standard" for evaluation, LLMs can serve as judges. By prompting LLMs to compare the generated results from two different competing models, it is possible to quantify their relative performance according to a given standard (Zheng et al.2023b). To ensure fairness and objectivity in scoring, we used three of the most advanced LLMs in the industry as evaluation models: GPT4-o1 [Hurst *et al.*, 2024], Deepseek-R1 [Guo *et al.*, 2025], and Gemini 2.5 Pro [Comanici *et al.*, 2025]. Each model independently scored the reasoning outputs from all compared methods, and "Total Score" was calculated as the average of the three. The scoring template was carefully designed and evaluates multiple aspects of the responses. The full template is provided in the Appendix Table 6. Furthermore, we invited experts to conduct a blind selection of the output results from each model, in order to ensure the consistency between human and machine evaluations. In addition, for oMeBench, we followed the original paper and conducted experiments exclusively on the oMeGold dataset. The evaluation method also employed the dynamic evaluation framework mentioned in the original paper, namely oMeS. It no longer relies solely on LLM for evaluation, but rather demonstrates the superiority of our method from a more objective perspective.

Knowledge Graph Construction

For our own dataset, we adopt a focused strategy for knowledge graph construction. Only reactants and products are used as nodes, while all other related entities are modeled as attributes on the edges between nodes. Each node is represented by the SMILES representation of the corresponding reactant or product, which is standardized using RDKit [Landrum, 2023]. Compounds that cannot be successfully standardized are filtered out during graph construction. This structure better captures the condition-centered relationships in chemical reactions. To support historical retrieval, we added historical information to the edge attributes. In the end, we built a chemical knowledge graph with about 130,000 nodes and 210,000 edges. The graph supports chain reasoning and efficient retrieval of multi-step reaction paths, and is stored in a Neo4j database. The detailed format of the knowledge graph is provided in the Appendix A.3.

In addition, for the oMeBench, we adopted the same construction approach as described above, using substance names as nodes and modeling detailed reaction mechanism information as edge condition attributes. During the graph construction process, we jointly used the oMeSilver, oMeTemplate, and oMeGold datasets to avoid oversimplification of the graph structure constructed solely based on the oMeGold data, which could lead to an overly easy retrieval process and subsequently affect the objectivity and credibility of experimental evaluation results.

Training Setup

In this experiment, we froze the parameters of the LLM and trained only the GNN component. The detailed training hyperparameters and GNN architecture are provided in the Appendix A.5.

Baseline Models

To evaluate our method's robustness across different model architectures, we selected LLaMA-3.1-8B [Dubey *et al.*, 2024] and Qwen2.5-7B [Qwen *et al.*, 2025] as baseline models. To compare our approach with larger open-source and closed-source models, we included Qwen2.5-72B and Deepseek-V3 [Liu *et al.*, 2025a] in the evaluation. To analyze performance differences between our method and deep reasoning models, we compared with Qwen3-8B. In addition, we compared REKG-MCTS [Song *et al.*, 2025], a state-of-the-art baseline for LLM reasoning in knowledge graphs.

4.2 Experimental Result

Our Test Set

Table 1 presents the comparison of model scores on our test set. Average represents the mean of score from the three evaluation models and is used to reflect the overall performance. According to the experimental results, LLaMA-3.1-

IJCAI-ECAI 2026 PREPRINT

Model	Size	Overall				Easy				Medium				Hard			
		V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}
Proprietary and Frontier Models																	
† GPT-5	–	90.4	51.9	23.6	29.1	92.4	69.9	42.5	51.8	90.0	50.7	21.6	26.5	**90.2**	31.5	7.2	10.0
† GPT-4o	–	83.7	20.3	3.9	5.0	82.6	26.0	8.3	10.1	83.8	20.6	3.2	4.2	85.1	9.7	1.6	1.6
† Claude-Sonnet-4	–	**91.0**	37.3	14.9	17.9	93.0	54.3	28.9	33.9	**90.9**	35.8	12.7	15.6	88.7	20.7	7.5	7.9
† Gemini-Pro-2.5	–	90.5	57.8	34.9	37.9	**94.5**	70.5	56.5	61.3	90.3	56.6	32.4	35.2	85.4	46.3	16.6	18.5
† DeepSeek-R1*	685B	87.6	45.4	22.5	25.0	88.4	62.3	39.0	41.9	87.9	45.1	20.6	23.3	82.9	21.3	8.3	9.7
Open Source Models																	
† GPT-OSS	20B	78.2	31.5	12.9	14.2	88.4	47.6	25.1	28.0	77.0	30.3	11.7	12.6	70.2	14.8	1.8	2.4
LLaMA3.1	8B	63.7	12.3	1.7	2.5	65.8	12.4	3.1	5.3	63.0	12.8	1.5	2.1	64.4	9.7	0.7	0.9
Qwen2.5	7B	61.3	13.3	2.8	3.6	64.3	17.2	3.9	6.6	58.4	13.5	2.5	3.0	73.6	6.5	2.4	2.9
Knowledge graph Retrieval-augmented generation																	
LLaMA3.1-ChemKGL	8B	80.3	**78.6**	**64.0**	**67.0**	81.7	**81.2**	**62.5**	**69.9**	81.4	**80.1**	**66.7**	**68.9**	71.2	**66.1**	**50.0**	**51.3**
LLaMA3.1-REKG-MCTS	8B	72.5	63.2	47.5	51.3	69.2	60.3	48.3	51.6	74.2	71.7	59.8	50.8	65.4	62.7	44.2	46.3
Qwen2.5-ChemKGL	7B	76.4	57.0	38.9	41.1	65.9	39.0	29.8	31.4	77.1	60.5	40.2	42.6	88.5	63.8	45.0	46.7
Qwen2.5-REKG-MCTS	7B	69.3	49.2	23.2	28.1	53.2	22.4	21.2	22.6	66.3	51.8	34.4	29.7	73.5	60.9	43.1	44.9

Table 2: Experimental results for oMeBench. The best results are in **bold** and the second-best are underlined.The symbol † indicates that the experimental data of the model is directly extracted from the original paper.

8B and Qwen2.5-7B achieved significant improvements after incorporating ChemKGL, with score increases of 43.28 and 32.89, respectively. This demonstrates not only the effectiveness of ChemKGL in enhancing base models but also its robustness across different model architectures. Notably, the ChemKGL-enhanced versions of LLaMA-3.1-8B and Qwen2.5-7B outperformed larger models like Qwen2.5-72B and Deepseek-V3, showing that ChemKGL can significantly boost multi-step reasoning performance while keeping the model relatively lightweight. On the other hand, models equipped with deep reasoning capabilities, such as Qwen3-8B, performed better than the base models, which indicates that deep reasoning paradigms can indeed benefit multi-step chemical reaction tasks. However, even so, Qwen2.5-7B combined with ChemKGL still outperforms Qwen3-8B, with a notable improvement of 26.5 points. This indicates that the graph enhancement mechanism introduced by ChemKGL is superior to the deep reasoning paradigm based on CoT in CMSRPIT. Furthermore, while the integration of REKG-MCTS also leads to noticeable performance improvements, a clear performance gap remains when compared to ChemKGL. This indicates that although REKG-MCTS-based methods can enhance knowledge graph reasoning to some extent, ChemKGL provides a more effective and comprehensive utilization of structured chemical knowledge. Consequently, our proposed approach maintains a clear advantage over existing knowledge graph reasoning baselines in addressing complex multi-step chemical reasoning tasks. Furthermore, Appendix Table 7 reports additional experimental analyses based on a larger test set. We evaluated responses with 2 to 5 reasoning steps by averaging the scores from three evaluation models. The results reveal a clear downward trend in scores as the number of reasoning steps increases, indicating that performance degrades with growing task complexity. This trend aligns with intuitive expectations and provides indirect evidence for the reliability and effectiveness of our evaluation method in capturing performance variations across different levels of reasoning complexity.

IJCAI-ECAI 2026 PREPRINT

Expert blind selection

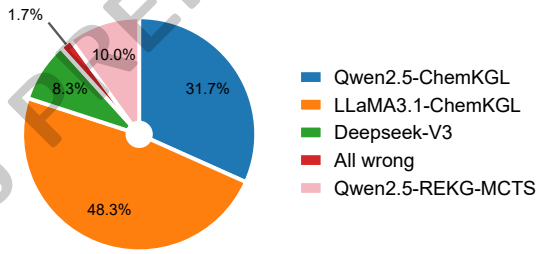

Figure 3: Expert blind selection results. All wrong means that the output results of all models are unreasonable and the best answer cannot be selected.

To further improve the credibility of the evaluation results, we invited chemistry experts to conduct a blind evaluation of the outputs generated by the nine models in Table 1. For each test case, experts compared the outputs of the nine models without knowing their identities. They selected the best-performing result and provided specific reasons for their choices. Representative examples of the experts' rationales are presented in Appendix Table 8, while the complete selection criteria and detailed explanations are available in our code repository.

The aggregated results of the expert selections are shown as a pie chart in Figure 3. As illustrated in the figure, ChemKGL was selected as the best result in 80% of the test cases. This finding is highly consistent with the LLM-based automatic evaluation results. It demonstrates strong agreement between expert judgment and model-based evaluation, further confirming the reliability of the evaluation method and the credibility of the results.

oMeBench

Table 2 compares the model scores on oMeBench. The experimental results show that LLaMA-3.1-8B and Qwen2.5-7B still achieved significant improvements after incorporat-

Model Name	o1	R1	Gemini	Average
LLaMA3.1-ChemKGL	84.51	58.51	69.19	**70.74**
LLaMA3.1-no-gnn	81.19	53.78	61.48	65.48
LLaMA3.1-EBP	66.72	42.32	40.85	49.96
LLaMA3.1-one-shot	51.89	27.18	12.88	30.65
LLaMA3.1	45.40	26.74	10.23	27.46
Qwen2.5-ChemKGL	83.04	54.99	64.02	**67.35**
Qwen2.5-no-gnn	80.88	51.13	59.98	64.00
Qwen2.5-EBP	67.32	43.31	46.58	52.40
Qwen2.5-one-shot	59.90	29.32	12.48	33.90
Qwen2.5	59.41	30.68	13.28	34.46

Table 3: Experimental results for ablation experiment. Where o1 means GPT4-o1, R1 means Deepseek-R1, Gemini means Gemini 2.5Pro and EBP means Entropy-Based Pruning. The best results are in **bold** and the second-best are underlined.

Search method	Search accuracy(%)
Forward retrieval(BW = 1)	76.67
Forward retrieval(BW = 3)	91.70
Historical retrieval(BW = 3)	95.00
Reverse retrieval(BW = 3)	99.17
Fusion retrieval(BW = 3)	**100.00**

Table 4: The retrieval accuracy of different retrieval methods. Where BW means beam-width.

ing ChemKGL, and remained superior to the approach combining REKG-MCTS. Furthermore, on the three indicators of L,S_{tot},S_{part}, they even surpassed excellent closed-source models such as Gemini-Pro-2.5 and GPT-5. This indicates that incorporating ChemKGL can significantly enhance the model's output performance in terms of reaction mechanisms and reaction intermediates.

4.3 Ablation experiment

To evaluate the impact of different modules and strategies on model performance, we conducted ablation experiments. We explored three main questions: whether to introduce a GNN module, whether to use an entropy-based pruning strategy, and whether to adopt a hybrid retrieval strategy.

Use of the GNN module

We first examined the necessity of the GNN module. In Table 3, "LLaMA3.1-no-gnn" represents the model without the GNN module, and its performance dropped by 5.26 points compared to the full model. Similarly, "Qwen2.5-7B-no-gnn" sees a 3.35-point decrease. This shows that the GNN module is crucial; the structured information from subgraphs supports better reasoning. The results are shown in Table 3.

Use of entropy-based pruning

To test the importance of entropy-based pruning, we created an extreme scenario by removing the target product node for each question in the test set from the knowledge graph. This forces target-based retrieval to fail. Table 3 shows that the LLaMA3.1-EBP model, which uses entropy-based pruning, improved by 22.5 points over the original inference model (LLaMA3.1-8B). Qwen2.5-7B-EBP also improves by 17.94 points. These results prove that the pruning strategy is highly effective in retrieval failure cases, especially in cold-start scenarios where inference performance tends to drop.

To ensure the improvement does not simply come from the presence of answer format rather than meaningful content, we conducted a one-shot control experiment. Specifically, we randomly selected answer labels from unrelated questions and used them as one-shot examples. These examples are formally valid but semantically irrelevant to the current question. Results show that the one-shot setting did not lead to performance improvement. In fact, Qwen2.5-one-shot showed a slight drop. This indicates that formal hints alone cannot compensate for the lack of semantic information, and irrelevant or redundant prompts may even hinder model reasoning.

Use of hybrid retrieval strategy

Table 4 shows the accuracy of different retrieval strategies. We define a retrieval as successful when the starting and ending nodes of the retrieved subgraph exactly match the reactant and product in the question.

In forward retrieval, applying beam search (beam width = 3) raised the accuracy to 91.7%, significantly higher than 76.67% without beam search. This indicates that beam search plays a key role in improving retrieval quality. Adding historical information further boosted accuracy to 95%, showing that reference to past information helps the model make better retrieval decisions. Reverse retrieval performed even better, achieving an accuracy of 99.17%, but a few samples still failed in retrieval. To address this, we introduced a hybrid strategy combining "reverse retrieval $\rightarrow$ forward retrieval $\rightarrow$ historical retrieval." This approach achieved a 100% retrieval success rate, with all 120 test samples successfully matched. The result verifies both the effectiveness and completeness of the hybrid retrieval strategy.

Since reverse retrieval covers the majority of cases, forward and historical retrieval steps are only used as fallback mechanisms. As a result, the additional computational overhead introduced by the hybrid retrieval strategy remains manageable. Experimental results on a test set of 120 cases show that knowledge graph retrieval takes approximately 12 seconds per instance on average, which is comparable to the inference time of current deep reasoning models and falls within an acceptable range.

5 Conclusion

We proposed a reasoning framework called ChemKGL, which was enhanced by knowledge graph retrieval. The framework employed a hybrid retrieval strategy based on the principle of maximizing mutual information. It also introduced an entropy-based pruning mechanism to effectively alleviate the cold-start problem. In addition, ChemKGL leveraged graph neural networks to deeply integrate textual information with structured knowledge, thereby improving reasoning capabilities. Experimental results showed that ChemKGL demonstrated strong robustness across various backbone model architectures. It significantly enhanced the model's multi-step reasoning ability, and it not only surpassed LLMs with larger parameter scales, but also outperformed various models with deep reasoning capabilities. Overall, ChemKGL provided an efficient and highly generalizable new paradigm for multi-step chemical reaction pathway reasoning, offering strong scalability and practical value.

IJCAI-ECAI 2026 PREPRINT

Acknowledgments

This research was partially supported by the National Science and Technology Major Project (Grant No. 2023ZD0121103), the National Natural Science Foundation of China (Grant No. 62376086), the Fundamental Research Funds for the Central Universities (JZ2025HGTG0289, PA2025IISL0114), the Anhui Province Science and Technology Innovation Project (Grant No. 202423k09020010), and the Strategic Priority Research Program of Chinese Academy of Sciences (XDA0490000).

References

[Bai *et al.*, 2025] Xuefeng Bai, Song He, Yi Li, Yabo Xie, Xin Zhang, Wenli Du, and Jian-Rong Li. Construction of a knowledge graph for framework material enabled by large language models and its application. *npj Computational Materials*, 11(1):51, 2025.

[Bhattacharya *et al.*, 2024] Debjyoti Bhattacharya, Harrison J Cassady, Michael A Hickner, and Wesley F Reinhart. Large language models as molecular design engines. *Journal of Chemical Information and Modeling*, 64(18):7086–7096, 2024.

[Chen *et al.*, 2020] Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: learning retrosynthetic planning with neural guided a* search. In *International conference on machine learning*, pages 1608–1616. PMLR, 2020.

[Chen *et al.*, 2025] Justin Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, et al. Reverse thinking makes llms stronger reasoners. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8611–8630, 2025.

[Comanici *et al.*, 2025] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024.

[Edge *et al.*, 2025] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[Edwards *et al.*, 2022] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, 2022.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[Huang *et al.*, 2023] Peixin Huang, Xiang Zhao, Minghao Hu, Zhen Tan, and Weidong Xiao. T 2-ner: At wo-stage span-based framework for unified named entity recognition with t emplates. *Transactions of the association for Computational Linguistics*, 11:1265–1282, 2023.

[Hurst *et al.*, 2024] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card, 2024.

[Jin *et al.*, 2024] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842, 2024.

[Kim *et al.*, 2021] Junsu Kim, Sungsoo Ahn, Hankook Lee, and Jinwoo Shin. Self-improved retrosynthetic planning. In *International Conference on Machine Learning*, pages 5486–5495. PMLR, 2021.

[Landrum, 2023] Gregory Landrum. Rdkit: Open-source cheminformatics. https://www.rdkit.org, 2023.

[Liu *et al.*, 2025a] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report, 2025.

[Liu *et al.*, 2025b] Runxuan Liu, Luobei Luobei, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15269–15284, 2025.

[Lowe, 2012] Daniel M. Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, UK, 2012.

[Lu *et al.*, 2025] Haolang Lu, Yilian Liu, Jingxin Xu, Guoshun Nan, Yuanlong Yu, Zhican Chen, and Kun Wang. Auditing meta-cognitive hallucinations in reasoning large language models, 2025.

[M. Bran *et al.*, 2024] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

[Ma *et al.*, 2025] Qinyu Ma, Yuhao Zhou, and Jianfeng Li. Automated retrosynthesis planning of macromolecules using large language models and knowledge graphs. *Macromolecular Rapid Communications*, February 2025.

[Mavromatis and Karypis, 2025] Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for

IJCAI-ECAI 2026 PREPRINT

efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699, 2025.

[Medress *et al.*, 1977] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, Barry Ritea, et al. Speech understanding systems: Report of a steering committee. *Artificial Intelligence*, 9(3):307–316, 1977.

[Qwen *et al.*, 2025] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, et al. Qwen2.5 technical report, 2025.

[Shi *et al.*, 2024] Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. Retrieval-enhanced knowledge editing in language models for multi-hop question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2056–2066, 2024.

[Song *et al.*, 2025] Xiaozhuang Song, Shufei Zhang, and Tianshu Yu. Rekg-mcts: Reinforcing llm reasoning on knowledge graphs via training-free monte carlo tree search. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9288–9306, 2025.

[Tanneru *et al.*, 2024] Sree Harsha Tanneru, Dan Ley, Chirag Agarwal, and Himabindu Lakkaraju. On the hardness of faithful chain-of-thought reasoning in large language models, 2024.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[Wang *et al.*, 2025] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. Gpt-ner: Named entity recognition via large language models. In *Findings of the association for computational linguistics: NAACL 2025*, pages 4257–4275, 2025.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[Xie *et al.*, 2022] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

[Xu *et al.*, 2025] Ruiling Xu, Yifan Zhang, Qingyun Wang, Carl Edwards, and Heng Ji. omebench: Towards robust benchmarking of llms in organic mechanism elucidation and reasoning, 2025.

[Yadav and Bethard, 2018] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th international conference on computational linguistics*, pages 2145–2158, 2018.

[Yang *et al.*, 2019] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[Yasunaga *et al.*, 2021] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 535–546, 2021.

[Yu *et al.*, 2024] Kevin Yu, Jihye Roh, Ziang Li, Wenhao Gao, Runzhong Wang, and Connor W. Coley. Double-ended synthesis planning with goal-constrained bidirectional search. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 112919–112949. Curran Associates, Inc., 2024.

[Zhong *et al.*, 2024] Xianrui Zhong, Yufeng Du, Siru Ouyang, Ming Zhong, Tingfeng Luo, Qirong Ho, Hao Peng, Heng Ji, and Jiawei Han. Actionie: Action extraction from scientific literature with programming languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12656–12671, 2024.

[Zhou *et al.*, 2025] Yigeng Zhou, Wu Li, Yifan Lu, Jing Li, Fangming Liu, Meishan Zhang, Yequan Wang, Daojing He, Honghai Liu, and Min Zhang. Reflection on knowledge graph for large language models reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23840–23857, 2025.

IJCAI-ECAI 2026 PREPRINT