Journal of Artificial Intelligence Research, vol. 2, no.2, pp. 1-11, 2025

A Large-Model-Based Dual-Stage Reasoning QA Approach with Multi-Hop Path Selection

Xiangshi Li[a], Yupeng Gao[b], Zhe Zhan[a] and Hui Liu[c,*]

[a]School of Software, Dalian University of Technology, Dalian, China
[b]School of Computer Science and Technology, Dalian University of Technology, Dalian, China
[c]School of Foreign Languages, Dalian University of Technology, Dalian, China

ARTICLE INFO

Keywords:
knowledge reasoning
large language models
multi-hop question answering

ABSTRACT

Knowledge reasoning and question answering (QA) is significant in contemporary artificial intelligence. It simulates the logical reasoning processes of humans to derive new information from existing knowledge. However, current knowledge reasoning QA systems still face two challenges. On one hand, when dealing with multi-hop questions, the system needs to associate and reason across multiple pieces of knowledge paths, which not only increases computational complexity but also leads to inaccurate reasoning results. On the other hand, reasoning models often lack generalizability, they struggle to adapt to diverse real-world application contexts. These issues are particularly pronounced in industry and healthcare. For the industrial sector, equipment fault diagnosis and predictive maintenance require accurate understanding of complex technical terms and domain-specific jargon. These concepts are often highly specialized and abstract, making them difficult to reason. For healthcare domain, the high accuracy and reliability of QA systems are necessary. Any minor error in disease diagnosis could lead to serious consequences. Moreover, knowledge in these fields evolves rapidly, necessitating the system to update the knowledge base in real time. To address the aforementioned challenges, we propose a secondary reasoning QA algorithm enhanced by large language models for multi-hop questions. Our model leverages attention mechanisms to iteratively optimize path selection and utilizes the semantic understanding capabilities of LLM to further excavate the semantic background. Through experimental validation on four datasets, our model has demonstrated remarkable performance on multihop QA tasks, significantly enhancing the accuracy and generalization of the QA system.

1. Introduction

Multi-hop knowledge-based question answering is the task of answering a question in natural language by obtaining and composing two or more independent pieces of evidence through multiple reasoning steps[1].At their core, existing multihop knowledge-reasoning QA methods can be grouped into three paradigms: memory-augmented, graph-neural network, and retrieval-generation fusion. Within the memory-augmented paradigm, KVMemNN[2] was the first to bring key-value memory to QA, performing joint KB and text reasoning via multihop memory reads. SRN[3] explicitly incorporates a search-and-reason cycle within the memory network to mitigate cascading errors. With the advent of graph neural networks, GraftNet[4] simultaneously constructs graphs from KB sub-graphs and text passages, using GNNs to perform multi-granularity aggregation. PullNet[5] iteratively "pulls" relevant entities and sub-graphs, substantially enlarging the reasoning scope. ReifKB[6] restructures KB triples into hyperedges, leveraging hypergraph convolution to capture high-order relations. BINET[7] introduces a bidirectional information bottleneck to prune redundant KG neighborhoods, enhancing performance on 3-hop questions. Hic-KGQA[8] employs hierarchical contrastive learning to address the false negative supervision problem in multihopscenarios. In the retrieval-generation fusion paradigm, VRN[9] treats path search as latent-variable inference through a variational reasoning network. EmbedKGQA[10] learns entity and relation embeddings directly, eliminating the need for explicit paths and allowing inference via simple vector matching. CBR-SUBC[11] adopts case-based reasoning, rapidly adapting to new domains by reusing subgraphs.

Despite recent advances, multi-hop knowledge reasoning QA still confronts three key challenges:

1. **Semantic–Structure Gap:** heterogeneous representations of text and the knowledge graph hinder unified reasoning and cause accuracy to drop sharply beyond two hops.
2. **Noise and Incompleteness:** missing edges in the KG and redundant textual information easily mislead memory and graph-based models when applied to large corpora.
3. **Opacity of Embeddings:** many efficient embedding-based methods generate reasoning paths that remain uninterpretable to humans.

To bridge these gaps, large language models have emerged with their profound semantic understanding, ushering in a fresh opportunity to tackle multi-hop question answering.[12]. Through unsupervised pre-training on large volumes of textual data, LLMs have amassed an extensive repository

DOI: https://doi.org/10.70891/JAIR.2025.080001
ISSN of JAIR: 3078-5529

*Corresponding author
liuhui1126@dlut.edu.cn (H. Liu)

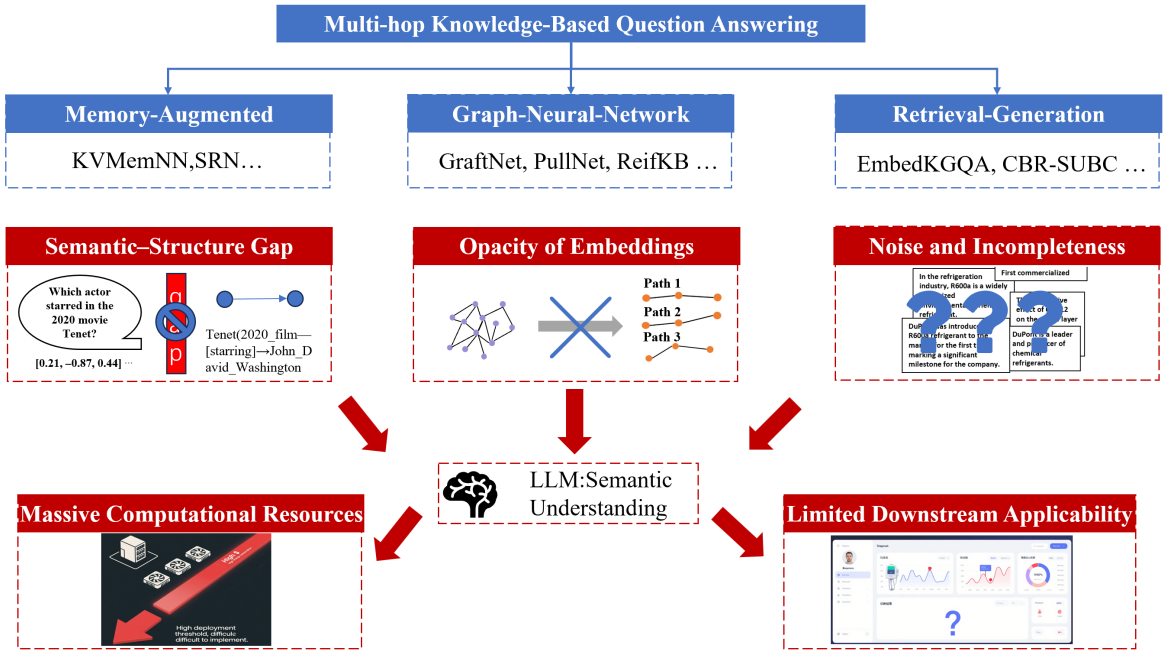
Multi-hop Knowledge-Based Question Answering
Memory-Augmented
KVMemNN,SRN…
Graph-Neural-Network
GraftNet, PullNet, ReifKB …
Retrieval-Generation
EmbedKGQA, CBR-SUBC …
Semantic–Structure Gap
Which actor starred in the 2020 movie Tenet?
[0.21, −0.87, 0.44] ···
Tenet(2020_film—[starring]→John_David_Washington
Opacity of Embeddings
Path 1
Path 2
Path 3
Noise and Incompleteness
LLM:Semantic Understanding
Massive Computational Resources
High deployment threshold, difficult to implement.
Limited Downstream Applicability

Figure 1: Current challenges in multi-hop knowledge-based question answering.

of semantic knowledge[13]. This enables them to more accurately comprehend the semantic nuances within questions and knowledge bases, thereby facilitating a more precise grasp of the key points and context of questions in multi-hop QA scenarios, which is essential for accurate reasoning. Their robust capability for multi-step reasoning, bolstered by complex neural network architectures and attention mechanisms, allows them to emulate human reasoning processes[14]. They can iteratively search for and integrate information within knowledge bases, effectively tackling the complex reasoning paths that are characteristic of multi-hop QA tasks. However, large language models still face three key challenges in multi-hop question answering: deep integration with knowledge graphs remains poorly interpretable; complex and abstract technical terms are hard to grasp, leading to weak downstream performance; and although semantic understanding is strong, computational costs are high, so efficiency must be improved[15].

Downstream tasks reveal the critical role of knowledge-reasoning QA across many domains. In industry and healthcare alike, it underpins fault diagnosis and clinical decisions. On the factory floor, models must predict equipment failure amid noisy data and jargon; one mistake can halt production[16]. In hospitals, clinicians rely on it to answer multi-hop questions from fast-evolving medical literature, where any error can harm patients. These high-stakes settings demand robust and interpretable question-answering systems.

In response to the existing challenges, We use attention to iteratively reason across steps and find the most likely paths. Because real tasks often involve many hops, we keep the top-three paths instead of the single best one. Their entity–relation triples are then fed to a large model for a second, redundancy-filtering pass, yielding more accurate answers. Our main contributions can be summarized as follows.

1. We propose a secondary reasoning question-answering algorithm enhanced by large language models, utilizing attention mechanisms specifically for multi-hop question answering.
2. We leverage the rich semantic understanding capabilities of large models to conduct secondary reasoning and correction on the top three ranked paths, thereby reducing the issue of errors caused by excessive similarity in model reasoning.
3. We have conducted extensive experiments on four datasets, and the results demonstrate that our proposed method exhibits strong generalizability and accuracy in multi-hop question answering.

2. Related Works

2.1. Multi-hop knowledge-based question answering

Early studies pointed out that the heterogeneous representations of text and KG are a performance bottleneck. KVMemNN[2] applied key–value memory for soft alignment, and SRN[3] later introduced an explicit search-and-reason loop, yet both remained at the level of vector-space mapping. To fully unify representations, KagNet[17] jointly trains text-relation extraction with KG-path scoring, while JointLK[18] couples text encoding and graph convolution via differentiable logic for end-to-end alignment. More recently, UniK-QA[19] proposes a "unified retrieval" framework that linearizes text, tables, and KG into a single sequence processed by one Transformer, significantly improving cross-modal three-hop accuracy. Although current work has gradually moved from alignment to unification, semantic understanding remains limited for corpora rich in complex and abstract terminology.

Large-scale knowledge graphs present twin challenges: exploding neighborhoods and noisy edges. GraftNet[4] and PullNet[5] adopt a "prune-then-expand" strategy, yet their thresholds remain manual. R-GSN[20] employs reinforcement learning to adaptively decide how many nodes to expand at each step, while SLEDGE[21] uses Monte-Carlo tree search to locate paths within milliseconds on 10-million-edge graphs. To combat noise, KG-BERT[22] first assigns confidence scores to triples before feeding them downstream, and CogQA[23] leverages heterogeneous graph attention to filter low-confidence edges. Nevertheless, these models still suffer from limited semantic understanding; as the number of hops grows, their performance drops, making accurate reasoning over complex, multi-turn questions elusive.

Traditional embedding methods lack human-readable explanations. ReifKB[6] visualizes high-order relations with hypergraphs, but interaction is limited. Path-based Attention Flow[24] maps GNN attention back to the original sentence, enabling a word-entity dual view. CausalKG-QA[25] introduces causal intervention to identify and remove spurious paths. Yet, current models generalize poorly: they perform well on specific datasets but still struggle to provide reasonable explanations when transferred to other domains.

2.2. Knowledge Reasoning with Large Models

With the emergence of large models, their powerful representation learning capabilities have inspired scholars both domestically and internationally to focus on methods that combine knowledge graphs with large pre-trained models for knowledge reasoning. For instance, KGATNs[26] introduce attention mechanisms to optimize the information propagation process within knowledge graphs, allowing models to focus on relationships or entities that are more critical to the task. R-GCN[27] integrates relational information into graph convolutional networks, capturing complex relationship features between entities through specially designed convolution operations. HAN[28] processes relationships and entities of multiple types using heterogeneous attention mechanisms, making it suitable for complex heterogeneous knowledge graphs. GraphBERT[29] extends the BERT model to graph-structured data, aggregating neighbor information through graph neural networks while retaining the pre-training advantages of BERT. K-Adapter[30] adapts pre-trained large models to knowledge graph data by adding adapter modules, enabling effective knowledge reasoning. The combination of knowledge graphs with large pre-trained models has demonstrated its strengths and potential in various aspects, but it still faces challenges in terms of computational resource consumption and data dependency.

In recent years, strong reasoning models became a research hotspot. Models like DeepSeek-R1[31] have achieved breakthroughs in mathematical reasoning and programming tasks, simulating human logical thinking and yielding excellent results. OpenAI's o1[32] model also excels in mathematical and logical reasoning tasks by introducing advanced reasoning mechanisms. To enhance the reasoning capabilities of large language models (LLMs), researchers have adopted three core strategies: prompt engineering, architectural optimization, and learning paradigm improvements. For example, in prompt engineering, Chain of Thought (CoT)[33] helps models break down complex problems into multiple smaller steps for step-by-step reasoning. In terms of architectural optimization, Retrieval-Augmented Generation (RAG)[34] enhances the model's memory capabilities, while Neuro-Symbolic AI[35] combines deep learning with symbolic logic to improve the reliability of reasoning. Currently, mainstream LLM reasoning frameworks are continuously being optimized, with frameworks such as XInference, LiteLLM, LMDeploy, SGLang, and vLLM constantly improving. Concurrently, the open-source infrastructure index provided by DeepSeek AI, including tools like FlashMLA, DeepEP, and DeepGEMM, offers underlying acceleration support for the performance enhancement of these reasoning frameworks. Reasoning tasks based on large models can leverage their advantages to achieve more accurate and reliable results.

3. Methodology

3.1. Attention-based Network

To address the multi-hop question answering tasks in diverse scenarios, we have developed an attention-based score propagation method. This approach enables the model to dynamically update and transfer the activation probabilities of entities across multiple steps by focusing on different parts of the question and computing the activation scores of relationships in the relation graph. Initially, the model initializes an entity score vector for the subject entity in the question, assigning it a score of 1 while setting the scores of all other entities to 0. In each step, the model attends to different segments of the question to identify the most relevant entity relationships. It then calculates the activation scores for each relationship in the graph, which are based on the current query context and propagate the entity scores from the previous step along the activated relationships to update the activation probabilities of the entities. This process is iterated until the required number of hops for the question is reached. Fig.1 illustrates the framework and the reasoning process.

Represent the entity scores at step t as a row vector $\mathrm{a}^t \in [0,1]^n$, where the values range between 0 and 1. a^0 is the initial score vector, with only the subject entity e^x having a value of 1. At step t, the query vector $\mathrm{q}^t \in \mathbb{R}^d$ is obtained, where d represents the hidden dimension.The

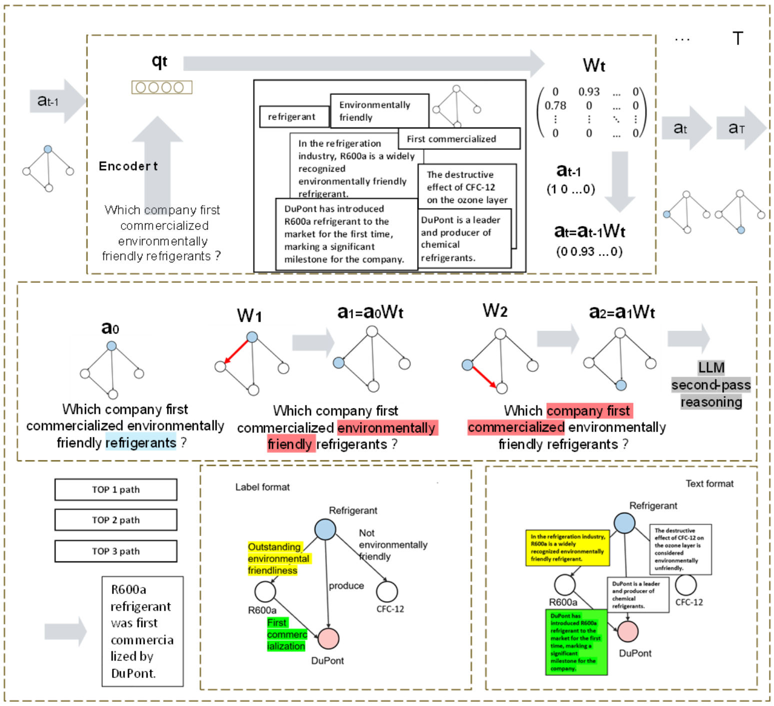
…
T
a_{t-1}
q_t
Encoder t
Which company first commercialized environmentally friendly refrigerants ?
refrigerant
Environmentally friendly
First commercialized
In the refrigeration industry, R600a is a widely recognized environmentally friendly refrigerant.
The destructive effect of CFC-12 on the ozone layer
DuPont has introduced R600a refrigerant to the market for the first time, marking a significant milestone for the company.
DuPont is a leader and producer of chemical refrigerants.
W_t
a_{t-1} (1 0 …0)
$a_t = a_{t-1}W_t$ (0 0.93 …0)
a_t
a_T
a_0
W_1
$a_1 = a_0 W_t$
W_2
$a_2 = a_1 W_t$
Which company first commercialized environmentally friendly refrigerants ?
LLM second-pass reasoning
TOP 1 path
TOP 2 path
TOP 3 path
R600a refrigerant was first commercialized by DuPont.
Label format
Refrigerant
Outstanding environmental friendliness
Not environmentally friendly
produce
R600a
CFC-12
First commercialization
DuPont
Text format
Refrigerant
In the refrigeration industry, R600a is a widely recognized environmentally friendly refrigerant.
The destructive effect of CFC-12 on the ozone layer is considered environmentally unfriendly.
R600a
DuPont is a leader and producer of chemical refrigerants.
CFC-12
DuPont has introduced R600a refrigerant to the market for the first time, marking a significant milestone for the company.
DuPont

Figure 2: The framework and inference process of TNLLM.

steps for calculating the score attention are shown in (1).

$$
\begin{aligned}
\mathrm{q}, (\mathrm{h}_1, \cdots, \mathrm{h}_{|q|}) &= \mathrm{Encoder}(q; \theta_e), \\
qk^t &= f^t(q; \theta_{f^t}), \\
b^t &= Softmax(qk^t \cdot [h_1; \cdots; h_{|q|}]^\top), \\
q^t &= \sum_{i=1}^{|q|} b_i^t h_i.
\end{aligned} \tag{1}
$$

where $\mathbf{q}$ is the embedding of the question, and f^t is the projection function at step t. It maps q to a specific query key qk^t, through which the attention score for each word's hidden vector h^i is calculated based on the query key.

As shown in (2), based on the query vector q^t, TNLLM computes the relationship score matrix $\mathrm{W}^t \in [0,1]^{n\times n}$, where θ_g is the learnable parameters. Different g need to be provided for label forms and text forms. This function outputs a matrix representing the transition probabilities between entities.

$$
\mathrm{W}^t = g(\mathrm{q}^t; \theta_g) \tag{2}
$$

Next, we simulate the process of skipping edges through (3), that is, by multiplying the entity score vector from the previous step with the current step's relationship score matrix to update the entity scores, as shown in (4).

$$
\mathrm{a}^t = \mathrm{a}^{t-1}\mathrm{W}^t \tag{3}
$$

$$
\mathrm{a}_j^t = \sum_{i=1}^{n} \mathrm{a}_i^{t-1} \times W_{i,j}^t \tag{4}
$$

Here, a^t represents the entity score vector at step t, W^t is the relationship score matrix at step t, and a_j^t is the score of entity j at step t.

After repeating t times, we obtain the entity scores for each step and then calculate their weighted sum as the final entity scores, as shown in (5).

$$
\begin{aligned}
\mathrm{c} &= \mathrm{Softmax}(\mathrm{MLP}(\mathrm{q})), \\
\mathrm{a}^* &= \sum_{t=1}^{T} c_t \mathrm{a}^t,
\end{aligned} \tag{5}
$$

where c is the distribution of the number of hops for the question, c_t is the probability of the t-th hop, and a* is the final entity score vector. It can automatically determine the number of hops to answer questions ranging from 1 hop to T hops.

The transparency and interpretability of the TNLLM model are among its key features. Through the aforementioned steps, it can reason about the relationships between entities to find answers.

3.2. Secondary Reasoning Enhanced By LLM

Relying solely on relationship scores to find subgraphs in a complex knowledge graph can lead to biases in multi-hop question answering. The complexity of relationships in the knowledge graph may cause important information to be omitted or irrelevant information to be included in the reasoning process when selecting subgraphs based solely on relationship scores. This bias may lead to inaccurate answers in the multi-hop reasoning process. To avoid such biases, secondary reasoning can be performed using an enhanced large model approach.

Firstly, the top 3 nodes with the highest scores are identified as shown in (6)

$$\{a_1^*, a_2^*, a_3^*\} = Top3(W_t) \tag{6}$$

Next, all corresponding path triples are found as shown in (7).

$$P = \{Find(a_0, a_1^*) \cup Find(a_0, a_2^*) \cup Find(a_0, a_3^*)\} \tag{7}$$

Finally, the triples are passed to the large model for judgment as shown in (8).

$$A = \text{LLM}(P, q, q^t) \tag{8}$$

3.3. Training Details

Based on the predefined standard answer set $Y = \{e_{y1}, \cdots, e_{y|Y|}\}$, construct the target score vector $y \in \{0,1\}^n$. For each entity e_i, if it is part of the standard answer set Y, the corresponding target score y_i is set to 1; otherwise, it is set to 0, as depicted in (9). This step ensures that the model has a clear optimization objective during the training process.

$$y_i = \begin{cases} 1, & \text{if } e_i \in Y, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The Euclidean distance between the entity score vector a^* and the target score vector $\mathbf{y}$ is utilized as the loss function $\mathcal{L}$ to guide the training process, as shown in (10).

$$\mathcal{L} = \|a^* - y\|. \tag{10}$$

As the TNLLM model is fully differentiable, it can effectively update model parameters to minimize the loss function using gradient descent and other optimization algorithms. This straightforward objective allows for learning all intermediate scores, including question attention, relation scores, and entity scores at each step.

3.4. Score Truncation And Language Masking

During the multi-hop reasoning process, as mentioned in (4), the entity score a_j^t may exceed 1. To avoid the problem of gradient explosion, we truncate the entity scores after each step to ensure they remain within the range [0,1], while also maintaining the differentiability of the operation. A truncation function is designed as shown in (11).

$$\text{Trunc}(a) = \frac{a}{z(a)}, \quad z(a) = \begin{cases} a.\text{detach}(), & \text{if } a > 1, \\ 1, & \text{if } a \leq 1. \end{cases} \tag{11}$$

To address potential answer ambiguity issues in text-based relationship graphs, a language masking mechanism is introduced. By predicting the masking scores for each entity and applying them to the final entity scores, the model's ability to capture key information in the question is enhanced. The question embedding is used to predict the masking scores for each entity. As shown in (12), where $m \in [0,1]^n$, m^i represents the masking score of entity e^i, and the multi-layer perceptron projects the d-dimensional features to n-dimensions.

$$m = \text{MLP}(q), \quad \hat{a}^* = m \odot a^* \tag{12}$$

3.5. Calculation Of Relationship Scores

In the TNLLM model, the calculation of relationship scores is one of the core steps, determining how the model reasons along the correct relationship paths during the multi-hop question answering process. This step involves matching the information in the question with relationships in the knowledge graph or textual corpus, thereby activating the most relevant entities and updating their scores. The relationship scores calculated in this paper include both labeled and textual forms.

In the labeled relationship graph, relationships are represented by a predefined vocabulary set $\mathcal{P}$. A multi-layer perceptron (MLP) is used to map the query vector $\mathbf{q}^t$ into the vocabulary space, and the Softmax function is applied to obtain the probability of each word as shown in (13).

$$p^t = \text{Softmax}(\text{MLP}(q^t)) \tag{13}$$

For each entity pair (e_i, e_j), according to its relationship label set $r_{i,j} = \{r_{i,j,1}, \cdots, r_{i,j,b}\}$, the corresponding word probabilities are collected as shown in (14), where b represents the maximum number of relationships between entity pairs.

$$W_{ij}^t = \sum_{k=1}^{b} p_{r_{i,j,k}}^t \tag{14}$$

In the textual relationship graph, relationships are described using natural language. The model first encodes each relationship description $r_{i,j,k}$ using an encoder to obtain its embedding representation, as shown in (15).

$$r_{i,j,k} = \text{Encoder}(r_{i,j,k}; \theta_r) \tag{15}$$

The relationship embedding is then element-wise multiplied with the query vector q^t, and the scores are calculated using an MLP and Sigmoid function as shown in (16).

$$p_t[r_{i,j,k}] = \text{Sigmoid}\left(\text{MLP}(r_{i,j,k} \odot q_t)\right) \tag{16}$$

For each entity pair, the scores of all relationship descriptions are summarized to obtain the final relationship score matrix W_t, as shown in (17).

$$W_t[i,j] = \sum_{k=1}^{b} p_t[r_{i,j,k}] \tag{17}$$

Due to the potentially vast relationship graph in practical applications, containing millions of relationship descriptions, directly computing the embedding and scores for all relationships is impractical. Therefore, this model adopts an efficient strategy, where in each step, only the entities with higher scores from the previous step are selected, and only relationships originating from these entities are considered. Additionally, if there are too many relationships that meet the condition, only the top ω relationships with the highest scores are retained.The pseudocode of our method is as follows.

```
Algorithm 1 TNLLM
Require: Relation graph $G$, which includes entity $\mathcal{E}$ and relation $\mathcal{R}$, multi-hop question $q$, subject entity $e_x$, answer enetity $Y = \{e_{y^1}, \dots, e_{y^{|Y|}}\}$, $f^t$ is the project function step $t$, problem hop count $T$, Truncation funciton Trunc($a$).
Ensure: The answer of question $A$.
 1: $q, (h_1, \dots, h_{|q|}) = \text{Encoder}(q; \theta_e)$, $a^0 = 1$
 2: for $t = 1$ to $T$ do
 3:     $qk^t = f^t(q; \theta_f)$
 4:     $b^t = \text{Softmax}\left(qk^t \cdot [h_i, \dots, h_{|q|}]^\top\right)$, $q^t = \sum_{i=1}^{|q|} b_i^t h_i$
 5:     $W^t = g(q^t; \theta_g)$, $a^t = a^{t-1} W^t$, $a_j^{t'} = \sum_{i=1}^{n} a_i^{t-1} \times W_{ij}^t$
 6:     $a_j^t = \text{Trunc}(a_j^{t'})$
 7: end for
 8: $c = \text{Softmax}(\text{MLP}(q))$
 9: $a^* = \sum_{i=1}^{n} c_t a^t$
10: $\mathcal{L} = \|a^* - y\|$
11: $m = \text{Sigmoid}(\text{MLP}(q))$, $\widehat{a^*} = m \odot a^*$
12: $\{a_1^*, a_2^*, a_3^*\} = \text{Top3}(W_t)$
13: $P = \left(\text{Find}(a_0, a_1^*) \cup \text{Find}(a_0, a_2^*) \cup \text{Find}(a_0, a_3^*)\right)$
14: $A = \text{LLM}(P, q, q^t)$
15: return $A$
```

4. Experiments

4.1. Datasets

We selected four datasets for our experiments, and the standards of the datasets are shown in Table 1.

1. **MetaQA**[36]: The dataset is expanded from the WikiMovies dataset and includes multi-hop questions. For 1-hop questions, there are 96,106 questions for the training set, 9,992 for the validation set, and 9,947 for the test set; for 2-hop questions, there are 118,945 questions for the training set, 14,872 for the validation set, and 14,872 for the test set; for 3-hop questions, there are 114,196 questions for the training set, 14,274 for the validation set, and 14,274 for the test set. Its knowledge graph comes from the movie domain, containing 43,000 entities, 9 predicates, and 135,000 triples. In addition to the labeled form, the textual form of the MetaQA dataset is extracted from the WikiMovies textual corpus.
2. **WebQSP**[37]: The dataset is based on the Freebase natural language question dataset. It includes 2,998 questions for the training set, 100 questions for the validation set, and 1,639 questions for the test set. The questions are either 1-hop or 2-hop in nature. The knowledge graph comprises 1.8 million entities, 572 predicates, and 5.7 million triples. Due to its large scale, the knowledge graph has been pruned.
3. **ComWebQ**[38]: The dataset is an expanded version of WebQSP, incorporating additional hops and constraints. It contains 27,623 questions for the training set, 3,518 questions for the validation set, and 3,531 questions for the test set. Subgraphs are retrieved for each question using the PageRank algorithm, with an average of 1,948 entities per subgraph and a recall rate of 64%.
4. **FalutBS:** The dataset utilizes a large model-generated industrial fault corpus, with 3,000 questions allocated for the training set, 314 questions for the validation set, and 314 questions for the test set. The knowledge graph is derived from a large-scale industrial knowledge base that has been constructed, comprising relevant materials from professional books, enterprise industry manuals, and online knowledge sources.

Table 1
Dataset Statistics

Dataset	Training Set	Test Set	Validation Set
MetaQA 1-hop	96,106	9,992	9,947
MetaQA 2-hop	118,948	14,872	14,872
MetaQA 3-hop	114,196	14,274	14,274
WebQSP	2,998	100	1,639
ComWebQ	27,623	3,518	3,531
FaultBS	3,000	314	314

4.2. Baseline Method

To thoroughly validate the effectiveness of our model, this paper selects 10 baseline methods, including KVMemNN [2], VRN[9], SRN[3], GraftNet[4], PullNet[5], ReifKB[6], EmbedKGQA[10], CBR-SUBC[11], BINET[7], Hic-KGQA [8]. The following is an introduction to several of these baseline methods.

1. **KVMemNN**: Utilizes a key-value memory network to store knowledge and performs multi-hop reasoning through iterative reading of memories.
2. **VRN**: The learner learns reasoning paths through reinforcement learning, with intermediate results that offer good interpretability.
3. **SRN**: Improves upon VRN by enhancing speed and performance through constrained search and reward shaping strategies.
4. **GraftNet**: Employs a heuristic method to extract subgraphs relevant to the question from the complete relationship graph and uses neural networks to infer answers.
5. **PullNet**: Builds upon GraftNet, learning to retrieve subgraphs through graph convolutional networks.
6. **ReifKB**: Proposes an expandable labeled knowledge base probability transmission method, which can be considered a simplified version of TransferNet.
7. **EmbedKGQA**: Treats knowledge graph question answering as a link prediction task and leverages knowledge graph embedding to assist in predicting answers.
8. **CBR-SUBC**: Employs core technologies of case-based reasoning and subgraph matching, identifying similar subgraph structures in historical cases to match the semantics and relationships of the current question, thereby generating answers.

9. **BINET**: Uses bidirectional reasoning and attention mechanisms as core technologies, combining forward and backward reasoning paths, starting from the subject entity of the question to gradually deduce possible answers; and starting from the candidate answers to verify their relevance to the question, optimizing the reasoning results through bidirectional interaction.
10. **Hic-KGQA**: Utilizes meta-paths in hypergraphs to model high-order relationships (such as multi-relational) and optimizes the multi-hop question answering process through reasoning chains.

4.3. Experimental Setup

1. **Model Parameters**: For MetaQA, the number of hops is set to 3, with a bidirectional GRU encoder for questions, featuring a hidden dimension of 1024. The relation encoder also employs a bidirectional GRU, with a threshold of 0.7 for selecting relations, considering up to 400 relations. For WebQSP, ComWebQ, and FalutBS, the number of hops is set to 2, with the question encoder pre-trained as BERT and fine-tuned for the task.
2. **Training Details**: The optimizer is Adam, with a learning rate of 0.001, and the training consists of 20 epochs. In terms of training time, on an NVIDIA 1080Ti GPU, the labeled format requires approximately one day, while the textual format takes slightly longer.
3. **Evaluation Metrics**: The primary evaluation metric is Hits@1 (the accuracy of top-1 predictions).

5. Results

5.1. Experimental Results of Labeled Formats

Table 2 presents the experimental results of various models on the labeled format of the MetaQA dataset. MetaQA is a large-scale multi-hop question answering dataset, comprising over 400,000 questions covering tasks up to three hops. Our method achieved an accuracy of 97.5% on 1-hop questions, which is comparable to four existing models; on 2-hop questions, it reached 100% accuracy, significantly outperforming other models; on 3-hop questions, it also achieved 100% accuracy, nearly solving all multi-hop question answering tasks on this dataset. Since, aside from the model proposed in this paper, four baseline models also achieved the highest accuracy, and the values are numerically identical, we analyzed the errors in their 1-hop questions. The errors stem from the ambiguity of entities. For instance, when asked about the lead actors of a movie, the knowledge graph might contain two movies with the same title but different release years. However, the standard answer provided by the MetaQA dataset only lists the 1920 version, leading to this unavoidable mismatch. Additionally, the two highest baseline models were similarly affected by the accuracy of the dataset. In the 2-hop and 3-hop stages, the ambiguity was eliminated through relational constraints.

Table 2
Hits@1 Experimental results MetaQA datasets

Methods	1-hop	2-hop	3-hop
KVMemNN	95.8	25.1	10.1
VRN	**97.5**	89.9	62.5
GraftNet	97.0	94.8	77.7
PullNet	97.0	99.9	91.4
SRN	97.0	95.1	75.2
ReifKB	96.2	81.1	72.3
EmbedKGQA	**97.5**	98.8	99.3
CBR-SUBC	**97.5**	99.8	99.2
BINET	97.0	99.8	99.0
Hic-KGQA	**97.5**	99.9	99.3
TNLLM(Ours)	**97.5**	**100.0**	**100.0**

Table 3
Hits@1 Experimental results on different datasets

Methods	WebQSP	ComWebQ	FalutBS
KVMemNN	46.7	21.1	34.7
GraftNet	66.4	32.8	39.8
PullNet	68.1	47.2	49.5
ReifKB	52.7	—	—
EmbedKGQA	66.6	—	44.8
CBR-SUBC	70.1	48.3	49.2
BINET	69.2	48.3	49.5
Hic-KGQA	70.8	50.9	49.7
TNLLM(Ours)	**71.5**	**51.1**	**63.8**

As the number of hops increases to two, performance differences among models begin to emerge. The accuracy of KVMemNN drops significantly to 25.1%, indicating its limited capability in handling multi-hop reasoning. In contrast, models such as PullNet, CBR-SUBC, BINET, Hic-KGQA, and TNLLM achieve accuracies around 99%, with TNLLM reaching 100%, demonstrating that these models have more effective mechanisms for multi-hop reasoning, capturing the associations between questions and entities and relationships within the knowledge graph. When it comes to three-hop questions, this disparity further widens, with TNLLM's accuracy still reaching 100%, suggesting that TNLLM can accurately find answers to complex multi-hop questions through a more rational architecture, more effective reasoning strategies, and a deeper understanding of the knowledge graph. The performance results of TNLLM and other baseline models on the WebQSP, ComWebQ, and FaultBS datasets are shown in Table 3. On the WebQSP dataset for two-hop questions, the accuracy reaches 71.5%, significantly outperforming the previous best models.

CompWebQ includes more hops and constraint conditions, achieving an accuracy of 50.8% on 2-hop questions, outperforming PullNet's 47.2%. Despite the increased complexity of CompWebQ's questions, TNLLM still demonstrates excellent performance, further proving its advantage in multi-hop question answering tasks. The Hic-KGQA

Table 4
Experimental results in textual form

Methods	MetaQA Text			MetaQA Text+50% Label		
	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
KVMemNN	75.4	7	19.5	75.7	48.4	35.2
GraftNet	82.5	36.2	40.2	91.5	69.5	66.4
PullNet	84.4	81.0	78.2	92.4	90.4	85.2
TNLLM (Ours)	**95.8**	**98.2**	**94.5**	**96.2**	**98.6**	**94.7**

method utilizes a hypergraph completion module to generate multi-sense embeddings of entities, complementing the implicit relationships in the knowledge graph. Hypergraphs allow a single edge to connect multiple entities, flexibly representing complex relationships, and combining entity embeddings with symbolic reasoning enhances the ability to handle long paths and complex constraints, yielding particularly advantageous results on CompWebQ. However, it also faces the significant challenge of high computational costs associated with hypergraph construction. Although TNLLM does not achieve the highest accuracy, its performance is still leading compared to other baseline models. On the FaultBS dataset, TNLLM's accuracy is also optimal compared to other baseline models, with a noticeable gap in performance, reflecting its ability to handle the challenges of professional terminology and complex entity relationship networks in refrigeration industrial scenarios. TNLLM can dynamically allocate weights through attention mechanisms, accurately capturing semantic information in specific questions and knowledge graphs, deeply understanding the question's meaning and related entity relationships, thus providing accurate basis for subsequent reasoning. In complex entity-relation networks, TNLLM first analyzes the question. Next, it plans a concise reasoning path on the fly. By skipping irrelevant nodes, it speeds up the search and Selects the answer closest to the facts.For instance, when dealing with refrigeration system fault diagnosis, it quickly locates entities and relationships related to the fault, reducing unnecessary search and computation. Combining previous results, this model not only enhances accuracy in specific domains but also demonstrates certain generalizability in industrial fields.

5.2. Experimental Results of Textual Formats

As shown in Table 4, TNLLM was compared with several models capable of handling textual relationship formats. TNLLM demonstrated superior performance over the other models, particularly on 2-hop and 3-hop questions, improving accuracy from 81.0% to 98.2% and from 78.2% to 94.5%, respectively. PullNet and GraftNet both infer answers by implicitly aggregating graph features, thus failing to provide intermediate relationship paths. This indicates that TNLLM is not only suitable for labeled relationship graphs but can also effectively process relationship graphs based on natural language descriptions.

To comprehensively explore the impact of integrating labeled and textual information on TNLLM's performance, this paper designed and conducted comparative experiments in mixed formats. Specifically, 50% of the labels were randomly sampled from the dataset and structured into triples, which were then integrated into the text-based relationship graph. When processing mixed data, predicates were parsed as word statements, and their semantic features were deeply mined and encoded using a relationship encoder, achieving effective integration of heterogeneous information. The experimental results showed that using a 50% label-mixed-text input strategy significantly improved TNLLM's key performance metrics compared to pure textual input formats. This result fully validates the model's excellent compatibility in handling multi-source heterogeneous information, proving its ability to efficiently integrate structured knowledge from labels with semantic information from text, thereby enhancing the model's performance in complex reasoning tasks.

5.3. Ablation Study

1. **Without Score Truncation**: After removing the score truncation module, the model's accuracy in labeled format dropped from 99.4% to 94.7%, and in textual format, it plummeted from 95.8% to 75.3%. In textual format, data complexity and uncertainty are higher, potentially leading to greater numerical fluctuations and an increased risk of gradient explosion. The score truncation module reasonably constrains scores to prevent excessive growth and mitigates gradient explosion issues, ensuring model training stability and effectiveness. Without this module, the model may become unstable and struggle to learn and predict accurately, resulting in a sharp decline in accuracy. This demonstrates that in textual format, the score truncation module is crucial for maintaining model performance.
2. **Without Language Masking**: After eliminating the language masking module, the model's accuracy in textual format plummeted from 95.8% to 62.1%. When processing textual relationships, the text contains a significant amount of ambiguous or polysemous information that can interfere with the model's judgment of the correct answer. The language masking module filters key information from the text and shields against factors that may lead to incorrect judgments, helping the model focus more accurately on content related to the correct answer. Without the language masking module, the model cannot effectively filter out incorrect answer-related information, and under the influence of numerous distractions, it struggles to accurately deduce the correct result, leading to a substantial drop in accuracy. This fully illustrates that the language masking module significantly enhances the model's accuracy when processing textual format data and is essential for model performance optimization.
3. **Without Auxiliary Loss**: After removing the auxiliary loss, the model's accuracy in labeled format dropped from 99.4% to 98.6%, and in textual format,

Table 5
Results of ablation experiment

Methods	Labeled Format	Textual Format
w/o score truncation	94.7	75.3
w/o language mask	—	62.1
w/o auxiliary loss	98.6	94.1
w/o LLM	95.2	78.2
TNLLM	**99.4**	**95.8**

it decreased from 95.8% to 94.7%. Auxiliary loss provides additional supervision during the model's learning process, guiding the model to focus more on key aspects of the reasoning path. For instance, in hop prediction helps the model better understand the depth and steps required for the question, allowing the model to more rationally plan reasoning paths when exploring entity relationships in the knowledge graph, avoiding ineffective or incorrect branches. Without auxiliary loss, the model lacks this extra guidance, leading to less precise learning of reasoning paths and consequently limited overall performance improvement, resulting in a certain degree of accuracy decline. This shows that auxiliary loss actively promotes better learning of reasoning paths and enhances model performance.

4. **Without Large Model Secondary Reasoning**: After removing the large model secondary reasoning step, the model's accuracy in labeled format dropped from 99.4% to 95.2%, and in textual format, it decreased from 95.8% to 78.2%. Large model secondary reasoning is not a redundant operation; it deeply verifies and optimizes the initial reasoning results. During the initial reasoning process, the model may produce inaccurate results due to various reasons (such as noise, local inaccuracies in the knowledge graph, etc.). Large model secondary reasoning, with its strong language understanding and reasoning capabilities, comprehensively reviews the initial results, supplements missing information, corrects erroneous reasoning, and thus derives more accurate and reliable answers. Without large model secondary reasoning, the model loses this important verification and optimization mechanism, significantly affecting accuracy and making it difficult to achieve higher levels. This clearly indicates that large model secondary reasoning plays an indispensable role in improving model accuracy.

In summary, each module of TNLLM plays a key role in enhancing the model's accuracy from different perspectives, working together to ensure the model's high performance.

5.4. Efficiency Experiment

We conducted experiments to verify the advantages of TNLLM in terms of data efficiency and convergence speed, demonstrating its ability to achieve excellent performance with less data and faster speeds during training. As shown in Figure 3, the x-axis represents the proportion of training data , and the y-axis represents the average Hits@1 accuracy. When using 10% of the training data, TNLLM's accuracy is almost identical to that achieved with the full training set (nearly 100%), while the accuracy of SRN and EmbedKGQA is significantly lower than that of TNLLM. This indicates that TNLLM can learn effective reasoning patterns with a small amount of data, reducing reliance on large-scale labeled data. As the proportion of training data increases, the TNLLM proposed in this paper consistently maintains high accuracy without overfitting or performance fluctuations, indicating its robust generalization capabilities due to model design elements such as attention mechanisms and score propagation.

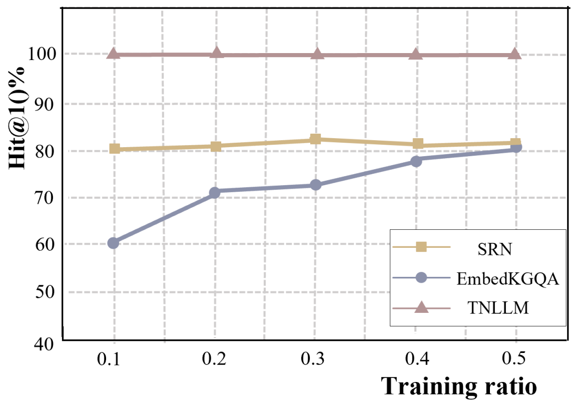

Figure 3: Chart of Experimental Results on Data Efficiency.

6. Conclusion

In response to the complexity and multi-hop characteristics of knowledge within specialized domains, we propose an enhanced secondary reasoning question-answering algorithm based on large language models. The algorithm works in two clear stages. First, it uses attention mechanisms to iteratively explore possible paths across multiple reasoning steps. Second, it leverages the rich semantic understanding of large models to perform a secondary round of reasoning on the top-three ranked paths. Together, these stages significantly improve the accuracy of multi-hop question answering. Extensive experiments on multiple datasets have validated the effectiveness of this approach. Our work is applicable to practical scenarios, particularly in industrial fault diagnosis and intelligent medical services in the healthcare sector—domains where the volume of knowledge is vast and accuracy is of utmost importance.

Future work can be expanded and deepened in several directions. First, we aim to further optimize the secondary reasoning algorithm to improve reasoning efficiency and accuracy. We will explore more advanced models and technologies, such as deep learning and natural language processing, to better address complex reasoning tasks in specialized domains. In addition, we plan to continuously expand and update the knowledge graph by incorporating more entities, relationships, and attributes. We will also consider integrating additional data sources, such as expert knowledge and industry standards, to enrich the content and structure of the knowledge graph.

References

[1] V. Mavi, A. Jangra, and A. Jatowt, "Multi-hop question answering," *Foundations and Trends® in Information Retrieval*, vol. 17, no. 5, pp. 457–586, 2024.

[2] A. H. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA, 2016, pp. 1400–1409.

[3] Y. Shen, P.-S. Huang, J. Gao, and W. Chen, "Reinforced self-attention network for multi-hop machine reading comprehension," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, USA, 2020, pp. 8889–8896.

[4] H. Sun, T. Bedrax-Weiss, and W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 4231–4242.

[5] H. Sun, M. Ma, W. Cohen, and C. T. Kwok, "PullNet: Open domain question answering with iterative retrieval on knowledge bases and text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019, pp. 2380–2390.

[6] D. Cohen, A. Yuan, and W. Cohen, "ReifKB: Reasoning with knowledge bases as hypergraphs," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 5952–5961.

[7] J. Tang, Y. Sun, Y. Lin, and X. He, "BINET: Bidirectional information bottleneck for knowledge graph reasoning," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xi'an, China, 2021, pp. 605–614.

[8] X. Zhang, Y. Li, D. Li, and Z. Zhang, "Hic-KGQA: Hierarchical contrastive learning for knowledge graph question answering," in *Proceedings of the 45th International ACM SIGIR Conference*, Madrid, Spain, 2022, pp. 1044–1053.

[9] M. Zhang, Y. Liu, H. Luan, M. Sun, T. Izuha, and J. Liu, "Variational reasoning network for multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 6069–6079.

[10] A. Saxena, A. Tripathi, and P. Talukdar, "EmbedKGQA: Improving multi-hop question answering over knowledge graphs using knowledge graph embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 4498–4507.

[11] R. Das, A. Zaheer, D. Thai, A. Godbole, E. Pereira, C. Tan, A. McCallum, and A. Anandkumar, "Case-based reasoning for better transfer in question answering," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, 2021, pp. 9359–9370.

[12] A. Chakraborty, "Multi-hop question answering over knowledge graphs using large language models," *arXiv preprint arXiv:2404.19234*, 2024. [Online]. Available: https://arxiv.org/abs/2404.19234

[13] Z.-Z. Li, D. Zhang, M.-L. Zhang, J. Zhang, Z. Liu, Y. Yao, H. Xu, J. Zheng, P.-J. Wang, X. Chen, et al., "From system 1 to system 2: A survey of reasoning large language models," *arXiv preprint arXiv:2502.17419*, 2025. [Online]. Available: https://arxiv.org/abs/2502.17419

[14] J. Wu and C. Cardie, "Reasoning Court: Combining Reasoning, Action, and Judgment for Multi-Hop Reasoning," *arXiv preprint arXiv:2504.09781*, 2025. [Online]. Available: https://arxiv.org/abs/2504.09781

[15] I. Barati, A. Ghafouri, and B. Minaei-Bidgoli, "Bactrainus: Optimizing Large Language Models for Multi-hop Complex Question Answering Tasks," *arXiv preprint arXiv:2501.06286*, 2025. [Online]. Available: https://arxiv.org/abs/2501.06286

[16] Q. Liu, S. Zhang, G. Qin, T. Ossowski, Y. Gu, Y. Jin, S. Kiblawi, S. Preston, M. Wei, P. Vozila, et al., "X-reasoner: Towards generalizable reasoning across modalities and domains," *arXiv preprint arXiv:2505.03981*, 2025. [Online]. Available: https://arxiv.org/abs/2505.03981

[17] H. Sun, T. Bedrax-Weiss, and W. W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 4231–4242.

[18] K. Sun, D. Yu, D. Yu, and C. Cardie, "JointLK: Joint reasoning with language models and knowledge graphs for question answering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Bangkok, Thailand, 2021, pp. 269–280.

[19] B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull, S. Gupta, Y. Mehdad, and S. Yih, "UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering," in *Findings of the Association for Computational Linguistics*, Seattle, WA, USA, 2022, pp. 1535–1546.

[20] Z. Wang, Y. Li, and J. Zhang, "R-GSN: Reinforced graph sampling and reasoning for large-scale knowledge graphs," in *Proceedings of the 34th Conference on Neural Information Processing Systems*, Vancouver, Canada, 2020, pp. 1–12.

[21] Y. Liu, X. Chen, and C. Zhang, "SLEDGE: Subgraph-guided Monte-Carlo tree search for large-scale knowledge graph reasoning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Virtual Event, 2021, pp. 1202–1212.

[22] X. Yao, J. Ye, Y. Lin, and M. Sun, "KG-BERT: BERT for knowledge graph completion and question answering," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019, pp. 4769–4779.

[23] M. Ding, C. Zhou, H. Yang, and J. Tang, "CogQA: Unifying knowledge graph and language understanding for multi-hop question answering," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 854–860.

[24] K. Ethayarajh and D. Jurafsky, "Attention flows are Shapley value explanations," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Bangkok, Thailand, 2021, pp. 1362–1371.

[25] J. Wu, X. Li, and W. Y. Wang, "CausalKG-QA: Causal intervention for robust multi-hop reasoning over knowledge graphs," in *Proceedings of the 10th International Conference on Learning Representations*, Virtual Event, 2022, pp. 1–15.

[26] W. Wang, et al., "KGAT: Knowledge Graph Attention Network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, AK, USA, 2019.

[27] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," in *European Semantic Web Conference*, Heraklion, Greece, 2018.

[28] Z. Zhang, P. Cui, X. Wang, et al., "Heterogeneous Graph Attention Network," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, 2019, pp. 2760–2766.

[29] W. Yu, et al., "GraphBERT: Only Attention is Needed," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020.

[30] W. Hu, et al., "K-Adapter: Flexible Transfer of Graph Neural Networks via Plug-and-Play Modules," in *Proceedings of the 35th Conference on Neural Information Processing Systems*, Online, 2021.

[31] D. Guo, D. Yang, H. Zhang, et al., "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[32] OpenAI Team, "o1: A New Frontier in AI Research," San Francisco: OpenAI, 2025.

[33] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[34] M. M. Abootorabi, A. Zobeiri, M. Dehghani, et al., "Ask in Any Modality: A Comprehensive Survey on Multimodal Retrieval-Augmented Generation," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

[35] B. C. Colelough and W. Regli, "Neuro-Symbolic AI in 2024: A Systematic Review," *CEUR Workshop Proceedings*, vol. 3819, pp. 1–15, 2024.

[36] Y. Zhang, H. Dai, Z. Kozareva, et al., "Variational reasoning for question answering with knowledge graph," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017.

[37] W.-t. Yih, M. Richardson, C. Meek, et al., "The value of semantic parse labeling for knowledge base question answering," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 201–206.

[38] A. Talmor and J. Berant, "The Web as a Knowledge-Base for Answering Complex Questions," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana, USA, 2018, pp. 641–651.