KELLM: Knowledge Graph Embedding Meets Large Language Models for Relation Prediction

Siyan Wu*
South China Normal University
Guangzhou, China
20233801098@m.scnu.edu.cn

Chenghua Zhu*
South China Normal University
Guangzhou, China
20233801014@m.scnu.edu.cn

Jieyu Zhan†
South China Normal University
Guangzhou, China
zhanjieyu@scnu.edu.cn

Lihua Cai†
South China Normal University
Guangzhou, China
lee.caip@m.scnu.edu.cn

Abstract

Knowledge graph completion (KGC) is a core task in database and AI systems, aiming to infer missing relations among entities to enhance structured knowledge utilization. However, existing methods struggle to jointly achieve structural reasoning, semantic precision, and computational efficiency. We present **KELLM** (**K**nowledge Graph **E**mbedding meets **L**arge **L**anguage **M**odels), a unified relation prediction framework that integrates symbolic embeddings, multi-hop reasoning, and large language models (LLMs). KELLM first leverages lightweight knowledge graph embeddings to efficiently retrieve high-recall candidate relations. A token translator module then maps continuous embeddings into discrete token representations, effectively bridging the structural and linguistic spaces and enabling seamless integration with LLMs. Multi-hop path descriptions are further incorporated as natural-language evidence to enhance complex reasoning and interpretability, while a semantic re-ranking stage exploits the global modeling capacity of LLMs to refine predictions. To ensure scalability, KELLM adopts parameter-efficient fine-tuning. Extensive experiments on FB15k-237, CoDEx-S, and DBpedia50 demonstrate that KELLM consistently outperforms embedding-based, pre-trained language model based, and LLM-based baselines, achieving substantial improvements in Mean Reciprocal Rank and Hits@K across all datasets. Our source code and datasets are available at our GitHub repository: https://github.com/TheBlueBanisters/KELLM.

Keywords

Knowledge Graph Completion, Knowledge Graph Embeddings, Large Language Models, Multi-hop Reasoning, Relation Prediction

*Equally contributed.
†Corresponding authors.

EDBT '26, Tampere, Finland

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Siyan Wu, Chenghua Zhu, Jieyu Zhan, and Lihua Cai. 2026. KELLM: Knowledge Graph Embedding Meets Large Language Models for Relation Prediction. In *Proceedings of 29th International Conference on Extending Database Technology (EDBT) (EDBT '26).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Knowledge graphs (KGs) have become essential data management infrastructure for representing entities and their relationships, supporting applications in search engines [1], recommendation systems [31], question answering [20], and scientific data management [6]. To address sparsity and incompleteness, knowledge graph completion (KGC) aims to infer missing facts from observed triples [2]. Within KGC, relation prediction, deciding the correct relation for a given entity pair, has become a central task [24].

Existing research on entity relation prediction in KGC tasks can be roughly divided into three paradigms. ***KGE Algorithms***, as illustrated in Fig. 1(a), follow a geometric or algebraic principle in which entities and relations are embedded into continuous vector spaces, and triples are scored by lightweight operators to achieve efficiency and scalability [2]. In typical behavior, subsequent formulations refine the representation of relational regularities while keeping inference economical [24]. A recurring limitation of many embedding-based approaches is that they rely primarily on structural signals, while ignore or weakly integrate ontology-level semantic constraints, which limits their ability to capture long-range or compositional patterns [6].

PLM-based Methods, as shown in Fig. 1(b), introduce textual semantics to complement sparse structural information. Textual descriptions or contextualized evidences enrich entity and relation representations when graph signals are insufficient [36]. Their typical behavior is to align language and knowledge spaces through unified pretraining, allowing textual cues to guide link prediction while maintaining compatibility with graph organization [34]. However, methods like KnowBERT integrate structured knowledge into language models through loose coupling (e.g., via entity linking and attention) rather than enforcing strict relational constraints, which may limit their ability to distinguish semantically similar relation types from heterogeneous ones in a knowledge-centric context [16].

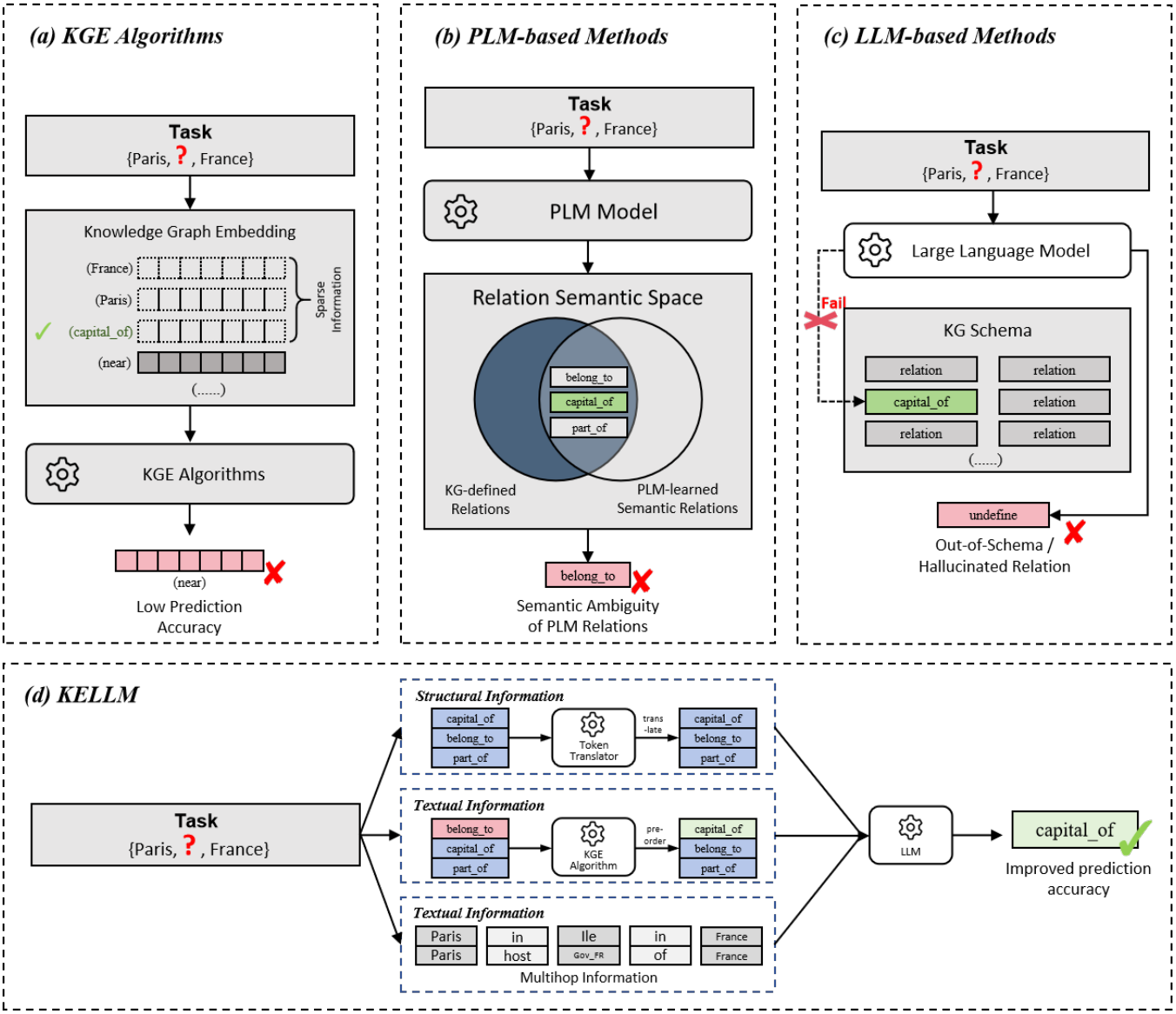

Figure 1: Conceptual comparison of existing paradigms for relation prediction. (a) *KGE Algorithms*: Knowledge Graph Embedding (KGE) models rely purely on structural information to infer relations, but often suffer from limited expressiveness and low prediction accuracy when encountering unseen or semantically ambiguous triples. (b) *PLM-based Methods*: Pre-trained Language Models (PLMs) introduce semantic knowledge into the reasoning process; however, their relational semantics may not align with the predefined schema of the knowledge graph, leading to semantic ambiguity and inconsistent mappings. (c) *LLM-based Methods*: Large Language Models (LLMs) exhibit strong generative reasoning ability but tend to produce out-of-schema or hallucinated relations when not grounded in the graph schema. This motivates the design of KELLM, which aims to unify structural constraints and semantic reasoning within a single framework. (d) Our proposed framework: *KELLM*.

LLM-based methods, as illustrated in Fig. 1(c), adopt a generative approach that infers relations from prompts, and retrieves contextual information to leverage broad, implicit knowledge [12]. In practice, using few-shot learning enables these models to synthesize or validate candidate relations under limited supervision, thus extending reasoning coverage in sparse relation representation space [13]. The main limitation of this paradigm is controllability: generations may deviate from the underlying schema and produce factual inconsistencies or hallucinations when responding to open-ended prompts [15].

Despite substantial progress, the aforementioned paradigms pose several significant challenges: (1) *Structural Semantic Disconnection.*

models often encode complementary facets in separate spaces, limiting unified reasoning [29]. Tighter composition across structural and semantic signals is required. (2) *Reasoning Consistency and Controllability.* Ensuring conformance to schema and logical constraints is difficult when prediction is generative or weakly structured [38]. We need to bridge the gap between generative models and KG structural constraints while leveraging LLM's powerful reasoning capabilities. (3) *Multi-hop Semantic Association.* Capturing cross-path and contextual dependencies remains challenging for complex and inductive relations [26]. Methods that incorporate multi-hop relational path information provide richer semantic cues and help mitigate these complicated cases.

To address these challenges, we propose **KELLM** (**K**nowledge Graph **E**mbedding meets **L**arge **L**anguage **M**odels), a unified relation prediction framework that integrates structural reasoning, semantic understanding, and efficient inference, as illustrated in Figure 1(d). KELLM first employs knowledge graph embedding models to encode structural patterns among entities and efficiently generate candidate relations, providing the model with explicit structural priors for reasoning. A lightweight *token translator* then maps continuous embeddings into discrete token representations, bridging the gap between symbolic structure and the language model's textual input space. To enhance higher-order reasoning and interpretability, KELLM incorporates multi-hop relational paths as natural-language descriptions, allowing the model to capture long-range dependencies that are often missed by embedding-based approaches. Finally, KELLM performs semantic re-ranking under structural guidance, combining the relational inductive bias of embeddings with the contextual reasoning ability of LLMs, thus achieving a better balance between reasoning accuracy, semantic coherence, and computational efficiency. Our main contributions are summarized as follows:

- We propose a unified reasoning framework that integrates structural and textual knowledge, enabling language models to directly utilize KG embeddings for semantic reasoning and thereby improving prediction accuracy while maintaining computational efficiency.
- We model several key mechanisms within the framework, including a Token Translator that maps structural vectors into language-understandable symbols, a two-stage reasoning mechanism for candidate filtering and semantic re-ranking, and a structure–semantic fusion strategy that explicitly leverages multi-hop path information. These mechanisms work collaboratively to enhance semantic modeling capability and interpretability.
- We conduct extensive experiments on multiple public datasets, including FB15k-237, CoDEx-S, and DBpedia50. Results show that our proposed framework, KELLM, significantly outperforms existing baselines on MRR and Hits@k, and ablation studies further verify the effectiveness of each module in it.

2 Related Work

Knowledge graph completion has been extensively studied from three complementary perspectives: structural embedding, language-enhanced modeling, and large language model–based reasoning. Early embedding approaches emphasize efficient structural representation but lack semantic expressiveness; subsequent PLM-based methods enrich textual semantics yet remain limited in structural reasoning and scalability; and recent LLM-based paradigms exploit powerful generative reasoning but still struggle to align symbolic constraints with semantic flexibility.

2.1 Knowledge Graph Embedding Methods

Knowledge Graph Embedding (KGE) methods learn low-dimensional representations for entities and relations, and employ scoring functions to measure the plausibility of triples. Distance-based models, such as TransE [2], interpret a relation as a translation between head and tail embeddings, favoring one-to-one or hierarchical patterns while remaining computationally efficient. Tensor factorization-based approaches capture multiplicative interactions between entity and relation vectors, with representative examples including DistMult [37] and its complex-valued variant ComplEx [28]. Geometry- and rotation-based models, exemplified by RotatE [24], encode relations as rotations in the complex plane to handle symmetry, anti-symmetry, and composition with clear geometric regularities. While these methods are scalable and structurally consistent, they mainly rely on topological signals, which limits their ability to capture implicit semantics and perform long-range compositional reasoning in sparse graphs.

2.2 Pre-trained Language Model-based Methods

To alleviate the limitations of purely structural embeddings, pre-trained language models (PLMs) have been leveraged to enrich knowledge graph representations with textual semantics. One research line encodes entity or relation descriptions using general-purpose encoders and fuses the resulting semantic features with structural embeddings. Representative backbones include BERT for contextualized token-level semantics [5], ALBERT for parameter-efficient pretraining [9], and ELECTRA for discriminator-style pre-training [4]. Another line of work strengthens the interaction between language and structure through joint objectives or prompt-based supervision, allowing textual cues to directly inform link prediction. For instance, KEPLER [34] unifies knowledge embedding and language representation, CoLAKE [22] injects KG-aware contexts during pretraining, and SimKGC [33] employs contrastive alignment to enhance semantic discrimination. Knowledge-augmented encoders such as KnowBERT [16] further integrate external symbols or memory modules into PLMs to better ground factual semantics. Task-oriented architectures, including structure-augmented text modeling [30] and sequence-to-sequence formulations for completion and question answering [20], explicitly tailor the text–structure interface for reasoning tasks. Overall, PLM-based methods substantially improve semantic generalization compared with pure embedding models, but they still exhibit limited structural reasoning ability, imperfect schema alignment, and high computational cost when scaled to large graphs.

2.3 Large Language Model-based Methods

With the rapid advancement of large language models (LLMs), their strong parametric knowledge retention and emergent reasoning capabilities have opened new possibilities for relation prediction,

enabling multi-step inference with minimal task-specific supervision [3]. Despite these advantages, current LLMs often struggle to explicitly encode graph constraints or control computation when applied to large-scale knowledge graphs.

(1) Prompt-based and in-context learning methods. These approaches design prompts or in-context exemplars that guide LLMs to infer relations from internal knowledge or retrieved evidence. Probing studies show that language models memorize considerable factual knowledge that can be elicited through cloze-style templates for link prediction [17]. Retrieval-augmented prompting further injects external evidence during inference to enhance factuality and robustness for knowledge-intensive reasoning [11]. Although flexible and annotation-efficient, such methods are highly sensitive to prompt formulation and often lack explicit structural constraints under predefined schemas, leading to inconsistency on complex graphs.

(2) Parameter-efficient adaptation and instruction fine-tuning methods. These methods adapt LLMs for completion through efficient tuning and graph-aware supervision. Prefix or continuous prompting attaches a small set of trainable vectors to steer generation while freezing most parameters [14]; prompt tuning generalizes this idea across tasks with minimal trainable parameters [10]; and low-rank adaptation reduces training cost while preserving reasoning accuracy [7]. In parallel, graph-centered instruction tuning incorporates path or schema signals to improve structural awareness [32], while structure-aware interfaces make graph regularities more accessible to LLMs [8]. These approaches enhance stability and controllability compared with pure prompting, yet still struggle to balance structural fidelity, semantic precision, and computational efficiency.

In summary, LLM-based approaches broaden the design space of knowledge graph reasoning by combining open-domain semantics with structured inference. Yet they still struggle to align symbolic structure with generative semantics and remain inefficient for large-scale graphs. This highlights the need for lightweight, structure-aware integration between embedding representations and language models to enable efficient and interpretable reasoning.

3 Problem Definition

To establish a unified analytical framework, this section provides a formal definition of knowledge graphs, multi-hop semantic reasoning, and the relation prediction task.

3.1 Symbolic Representation of Knowledge Graphs

A knowledge graph (KG) can be formally defined as a collection of triples:

$$G = (E, R, T),$$

where E denotes the set of entities, R the set of relations, and $T \subseteq E \times R \times E$ the set of factual triples. Each triple $(h, r, t) \in T$ indicates that entity $h \in E$ is semantically connected to entity $t \in E$ via relation $r \in R$.

In practical database or knowledge system settings, T is typically incomplete, containing numerous potential but unrecorded facts $(h, r^*, t) \notin T$. Therefore, the central goal of knowledge graph reasoning is to estimate the plausibility of such missing relations r^*, and to complete the factual space $E \times R \times E$ based on the existing structure G.

3.2 Multi-hop Paths and Semantic Composition

The dependency between entities is not always represented by a single direct relation. Given an intermediate entity sequence $(e_1, e_2, \ldots, e_{k-1})$ and a corresponding relation sequence $(r_1, r_2, \ldots, r_k)$ satisfying

$$h \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \cdots \xrightarrow{r_k} t,$$

we define $p(h, t) = (r_1, r_2, \ldots, r_k)$ as a k-hop semantic path linking h and t. The path $p(h, t)$ forms a composite semantic relation, through which a latent relation r^* can be inferred as

$$(r_1, r_2, \ldots, r_k) \Rightarrow r^*.$$

Such multi-hop reasoning is common in scholarly knowledge graphs [23], recommendation systems [31], and causal-chain modeling [38]. Explicitly modeling multi-hop paths enables the system to capture higher-order semantics and enhances both reasoning ability and interpretability.

3.3 Formal Definition of Relation Prediction

The relation prediction task aims to determine the most plausible relation $r^* \in R$ between a given entity pair (h, t) under context C. Formally, a learnable scoring function is defined as:

$$f : E \times R \times E \times C \to \mathbb{R},$$

where $f(h, r, t, C)$ quantifies the plausibility of relation r linking the entity pair (h, t) given the contextual information C. The context C may include neighborhood structures, relational paths, or textual descriptions that provide complementary semantic cues.

The prediction process can then be expressed as:

$$r^* = \arg\max_{r \in R} f(h, r, t, C),$$

that is, selecting the relation with the highest score among all candidates. This definition characterizes the mapping from the entity–context space (h, t, C) to the optimal relation r^*. In essence, relation prediction can be viewed as a function learning problem over the symbolic space $E \times R \times E$, where the main challenge lies in effectively capturing high-order semantic interactions while achieving a balance between reasoning accuracy and computational scalability.

4 KELLM

KELLM is a unified framework for relation prediction that integrates structural representations from knowledge graph embeddings (KGEs) with the semantic reasoning ability of large language models (LLMs). The overall workflow proceeds sequentially from structural encoding to candidate filtering, followed by multi-hop semantic expansion and semantic-structural ranking, ultimately producing a ranked list of relations ordered by plausibility. Algorithm 1 illustrates the pseudo-codes of the KELLM framework.

4.1 Structural Encoding

As illustrated in Figure 2(a), the structural encoding module projects entities and relations of a knowledge graph into continuous vector

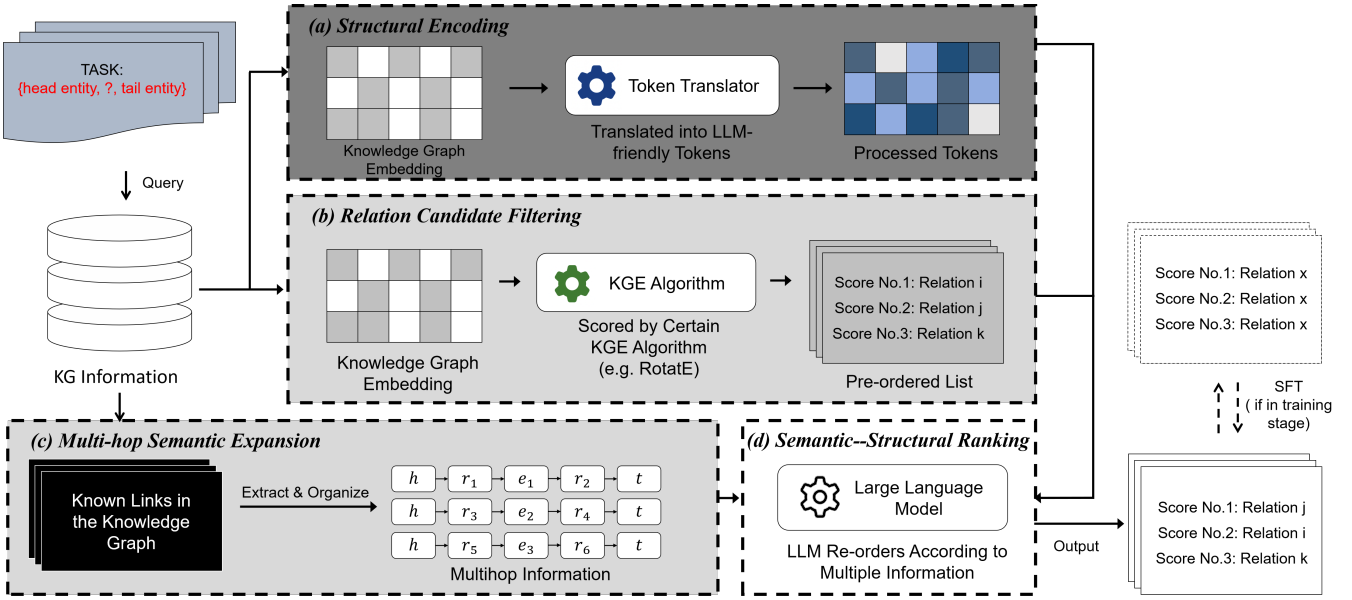

Figure 2: Overall architecture of the proposed KELLM framework for relation prediction. The framework integrates structural knowledge from knowledge graph embeddings (KGEs) with the semantic reasoning ability of large language models (LLMs). (a) *Structural Encoding*: entities and relations are projected into continuous vector spaces via KGE models and mapped into the LLM token space through a Token Translator. (b) *Relation Candidate Filtering*: KGE-based scoring generates a structural prior to select the top-k candidate relations. (c) *Multi-hop Semantic Expansion*: multi-hop paths are transformed into textualized semantic sequences that enrich contextual reasoning. (d) *Semantic–Structural Ranking*: the LLM integrates both structural and semantic representations to produce a ranked list of relations in descending order of plausibility.

spaces to capture latent structural dependencies. Formally, a KGE model [2, 24, 28, 37] is employed to generate dense embeddings:

$$\Phi_E : E \to \mathbb{R}^{d_e}, \quad \Phi_R : R \to \mathbb{R}^{d_r},$$

where d_e and d_r denote the embedding dimensions of entities and relations, respectively.

To align the structural embeddings with the semantic space of the language model, KELLM introduces a **token translator** module implemented as a multi-layer perceptron (MLP) [18]:

$$\phi(x) = \mathrm{MLP}(x),$$

which defines a nonlinear projection from the structural embedding space $\mathbb{R}^{d_{\text{in}}}$ to the token embedding space $\mathbb{R}^{d_{\text{LM}}}$ of the LLM. This design ensures both *dimensional* and *semantic alignment* between the two representational domains.

To further bridge the representation gap between continuous vectors and discrete language tokens, KELLM employs a quantization function:

$$z_x = \mathrm{Quantize}(\phi(x)), \quad z_x \in \mathbb{Z}^{d_{\text{LM}}},$$

where z_x denotes the discrete symbolic token corresponding to entity or relation x. The quantization operation discretizes the projected vector $\phi(x)$ by mapping it to the nearest token embedding in the LLM vocabulary or a learned codebook, effectively transforming structural knowledge into a symbol-compatible form that can be directly consumed by the language model.

In practice, this transformation is applied only to the query entity pair (h, t) and the top-k candidate relations selected during filtering, thereby maintaining computational efficiency while preserving representative structural information. The resulting symbolic vectors are subsequently integrated into the language model for joint reasoning over structure and semantics.

4.2 Relation Candidate Filtering

As shown in Figure 2(b), performing inference over the entire relation set R is computationally prohibitive for large-scale graphs. KELLM therefore constructs a structural prior through KGE-based scoring [2] to narrow down the candidate space:

$$S(h, r, t) = g\big(\Phi_E(h), \Phi_R(r), \Phi_E(t)\big),$$

where $g(\cdot)$ denotes the scoring function associated with the chosen KGE model (e.g., TransE[2] , DistMult[37], ComplEx[28] or RotatE[24]). Relations are ranked according to $S(h, r, t)$, and the top-k relations are retained:

$$R_{\text{cand}} = \mathrm{TopK}(S, k).$$

This filtering procedure substantially reduces computational overhead while preserving high recall, yielding a compact and informative candidate set that serves as the input for the subsequent ranking module.

Algorithm 1: Pseudo-code of the KELLM framework.

Algorithm 1: KELLM Framework for Relation Ranking

Input: Knowledge graph $G = (E, R, T)$, query entity pair (h, t), selected KGE model M

Output: Predicted relation r^*

// Compute structural scores for all relations.

$S(h, r, t) = g(\Phi_E(h), \Phi_R(r), \Phi_E(t)),\ \forall r \in R$

// Select top-k relations by score.

$R_{\text{cand}} = \textsc{TopK}(S, k)$

// Translate embeddings into discrete tokens.

$X \leftarrow \textsc{TokenTranslate}(h, R_{\text{cand}}, t)$

// Incorporate multi-hop paths into context.

for each path $p = (h, r_1, e_1, \ldots, r_m, t)$ **in** G **do**

  $X \leftarrow X \cup \textsc{EncodePath}(p)$

end for

// Differentiate between training and inference.

if training then

  $L = -\sum_i \log P_\theta(y_i | y_{<i}, X)$ // Compute sequence loss.

else

  $S(r) = M_\theta(X)$

  $r^* = \arg\max_{r \in R_{\text{cand}}} S(r)$

end if

4.3 Multi-hop Semantic Expansion

As illustrated in Figure 2(c), entity relations in real-world knowledge graphs often involve multi-hop reasoning paths rather than single direct links. To capture such higher-order semantic dependencies, KELLM extracts directed paths starting from the head entity h and ending at the tail entity t:

$$p = (h, r_1, e_1, r_2, e_2, \ldots, r_m, t),$$

where each e_i denotes an intermediate entity and r_i represents the relation connecting adjacent entities. This path encodes a reasoning chain that progressively links h to t through a sequence of semantic relations.

Since enumerating all possible paths over the entire graph is computationally infeasible, we follow the localized subgraph paradigm introduced in GraIL [26]. Specifically, we sample a k-hop subgraph centered at the head entity h, and enumerate simple relational chains within this subgraph that connect h to t. We further constrain the maximum path length (e.g., $m \leq 3$) and limit the number of candidate paths per entity pair to keep the extraction lightweight and scalable.

Each path p is then transformed into its natural-language representation $\psi(p)$, where $\psi(\cdot)$ denotes a function that maps a structured relational path into a textual description. For example:

> *"Entity h is connected to e_1 via relation r_1, further linked to e_2 through r_2, and finally reaches the target entity t."*

This textualized path representation abstracts structural reasoning into the linguistic space, allowing the language model to access structure-induced semantic cues within a unified representation

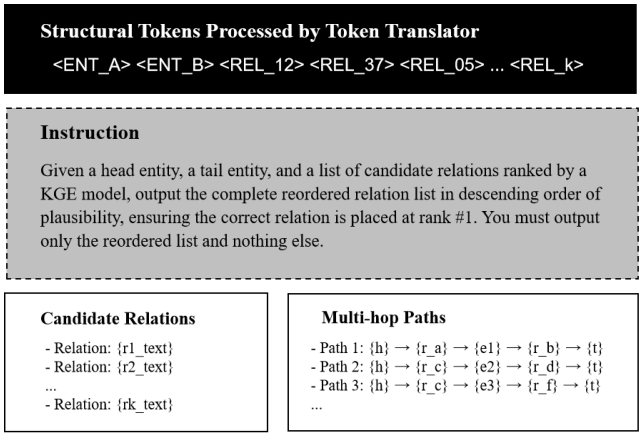

Figure 3: The standardized prompt template used in KELLM for relation ranking. Each prompt specifies the task instruction, query entity pair, and top-k candidate relations. The template provides consistent input formatting for the LLM, enabling fair comparison and stable ranking across queries.

framework. The final model input integrates symbolic embeddings and path textual information:

$$X = [z_h; z_{r_1}; \ldots; z_{r_m}; z_t; \psi(p)],$$

providing semantically grounded context for relation inference.

4.4 Semantic–Structural Ranking

As shown in Figure 2(d), once the candidate set and contextual information are prepared, KELLM formulates relation prediction as a fusion-based ranking task. Given input X, the language model M_θ computes contextual plausibility scores for all candidate relations:

$$S_{\text{LLM}}(r_i) = M_\theta(X, r_i).$$

The ranked relation list is then obtained by sorting these scores in descending order:

$$\mathcal{R}_{\text{pred}} = \text{Sort}_\downarrow\big(S_{\text{LLM}}(r_i)\big), \quad r_i \in R_{\text{cand}}.$$

To ensure consistent reasoning across structural and semantic modalities, KELLM employs a **standardized prompt template**, which is adapted from the Alpaca instruction-tuning format [25]. This template organizes task instructions, candidate relations, and contextual cues into a unified structure, as illustrated in Figure 3. It normalizes the reasoning interface between symbolic and textual inputs, reduces prompt sensitivity across graph instances, and enhances interpretability by explicitly separating task intent from graph-specific content. Such controlled prompting enables fair comparison among candidate relations, stabilizes ranking behavior during both training and inference, and promotes reproducibility and cross-graph generalization.

4.5 Training and Parameter Optimization

KELLM is optimized in a sequence-to-sequence (Seq2Seq) manner, where the model learns to generate the ordered relation sequence conditioned on the structural and semantic inputs. During training,

the model minimizes the negative log-likelihood of the normalized ranking distribution:

$$\mathcal{L} = -\sum_{r_i \in R_{\text{cand}}} y_i \log P_\theta(r_i \mid X),$$

where y_i denotes the normalized label distribution over candidate relations. This listwise objective aligns model optimization with ranking-oriented inference, encouraging the model to produce globally consistent and semantically coherent relation orderings.

To enhance parameter efficiency, we adopt the Low-Rank Adaptation (LoRA) strategy [7]:

$$W = W_0 + AB^\top, \quad \text{rank}(A, B) = r \ll \min(d, k),$$

where A and B are trainable low-rank matrices. This configuration reduces memory and computational overhead while maintaining strong semantic–structural alignment performance.

4.6 Complexity Analysis

The computational complexity of KELLM mainly arises from four components: candidate filtering, token translation, language model inference, and multi-hop semantic modeling. Let $|R|$, $|E|$, and $|T|$ denote the number of relations, entities, and triples, respectively; d the embedding dimension; L the input sequence length; and k the number of candidate relations.

The candidate filtering stage requires $O(|R|d)$ operations due to KGE-based scoring over all relations. After top-k selection, token translation introduces $O(kd)$ cost, which is negligible compared to language model inference. The LLM inference stage dominates the overall runtime with complexity $O(kLd)$, as it processes the contextualized sequence containing k relation candidates. Multi-hop semantic modeling contributes approximately $O(|E| + |T|)$ complexity in the worst case but is effectively reduced to near-linear scale through heuristic path pruning.

In practice, the combination of candidate restriction and path pruning keeps the overall computational cost tractable even for large-scale knowledge graphs. The framework thus achieves a favorable balance between scalability and reasoning accuracy, maintaining $O(kLd)$ inference complexity dominated by the LLM forward pass while ensuring efficient preprocessing and high-quality relational ranking.

5 Experiments

5.1 Experimental Settings

In this section, we present the overall settings of the main experiments, including the datasets, baselines, evaluation metrics, and training details. The experiments aim to systematically evaluate the performance and generalization ability of our proposed model on knowledge graphs of varying scales and complexities.

We conduct experiments on three public knowledge graph datasets: **FB15k-237**, **CoDEx-S**, and **DBpedia50**. These datasets differ in scale, relational diversity, and semantic richness, as summarized in Table 2, enabling comprehensive assessment of the model's reasoning capability and robustness.

- **FB15k-237**: A subset of Freebase containing 14,541 entities and 237 relations with approximately 310K triples[27]. Redundant inverse relations are removed to prevent information leakage, providing a more realistic evaluation of reasoning performance.
- **CoDEx-S**: The small version of the CoDEx series[19], consisting of 2,034 entities, 42 relations, and about 35K triples. Despite its small size, it covers diverse semantic categories and complex relational structures, testing the model's semantic reasoning ability.
- **DBpedia50**: A subset extracted from the DBpedia knowledge base[1] with 351 relations and thousands of entities. The unbalanced and long-tailed relation distribution makes it suitable for evaluating model robustness under sparse data conditions.

Table 2: Statistics of the datasets used in our experiments.

Dataset	#Entities	#Relations	Train	Valid	Test
FB15k-237	14,541	237	272,115	17,535	19,496
CoDEx-S	2,034	42	32,888	1,827	1,828
DBpedia50	24,638	351	35,077	4,384	10,969

To ensure a comprehensive comparison, we select twelve representative baseline methods covering three paradigms: structured embedding models, pre-trained language models, and large language model (LLM)-based approaches. Their core ideas are summarized below.

- **TransE** [2] is a classical knowledge graph embedding method that models each relation as a vector translation, aiming to satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. It performs well on simple one-to-one or hierarchical relations but struggles with symmetric, inverse, and compositional patterns.
- **DistMult** [37] is a bilinear model that scores triples via the Hadamard product $\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$. While efficient and compact, it is inherently symmetric, making it unsuitable for modeling antisymmetric relations.
- **ComplEx** [28] extends DistMult into the complex vector space by introducing real and imaginary components with conjugate interactions. This formulation allows it to represent both symmetric and anti-symmetric relations effectively.
- **RotatE** [24] is a geometry-inspired embedding model that represents relations as rotations in the complex plane. It can naturally capture symmetric, anti-symmetric, and composition patterns, offering both strong interpretability and robust generalization.
- **KGBERT** [39] is a pre-trained language model approach that linearizes triples into natural language sequences and fine-tunes a BERT-based encoder for relation prediction. It leverages rich semantic knowledge from pretraining to complement structural embeddings.
- **BERTRL** [40] is a BERT-based relational learning method that integrates graph context and adaptively retrieves relational paths during inference. It generalizes to unseen entities and provides interpretable reasoning through contextualized PLM embeddings.

Table 3: Main experiment results (MRR and Hits@k). Bold indicates the best result, underline indicates the second-best.

Method	FB15k-237				CoDEx-S				DBpedia50			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
KGE Methods												
TransE	0.3803	0.2041	0.3487	0.5068	0.3460	0.2200	0.4010	0.5940	0.0838	0.0714	0.0876	0.1063
ComplEx	0.2711	0.1865	0.2959	0.4444	0.2470	0.1500	0.3010	0.4640	0.0209	0.0491	0.0253	0.0252
DistMult	0.2593	0.1815	0.2289	0.4148	0.3370	0.2180	0.3910	0.5690	0.0507	0.0453	0.0530	0.0624
RotatE	0.3241	0.2314	0.3573	0.5903	0.4500	0.3410	0.5060	0.6530	0.1062	0.0511	0.1430	0.1958
PLM Methods												
KGBERT	0.3853	0.3189	0.4073	0.5094	0.5036	0.3243	0.6291	0.8829	0.2190	0.1079	0.2443	**0.4400**
BERTRL	0.4778	0.3333	0.4761	**0.7128**	0.7004	0.6078	0.7733	**0.8866**	0.1645	0.0806	0.1612	0.3548
LASS	0.1173	0.0951	0.1131	0.1273	0.4376	0.3775	0.4792	0.4999	0.1811	0.1387	0.1623	0.2972
LLM Methods												
KG-LLM	0.2572	0.1640	0.2220	0.4520	0.1932	0.1160	0.2030	0.3070	0.2160	0.1030	0.1730	0.3620
InstructGLM	0.2807	0.2000	0.2350	0.5090	0.1824	0.0580	0.2770	0.3230	0.1920	0.0930	0.1330	0.3827
StructGPT	0.1661	0.0290	0.2160	0.3660	0.1824	0.0580	0.2770	0.3230	0.1920	0.0930	0.1330	0.3920
IO Prompt	0.4952	0.454	0.543	0.573	0.6533	0.5762	0.6549	0.7564	0.1156	0.099	0.126	0.142
CoT Prompt	0.2657	0.252	0.275	0.289	0.627	0.645	0.658	0.677	0.107	0.1209	0.13	0.148
Ours (KELLM)	**0.5871**	**0.5550**	**0.6170**	0.6330	**0.7627**	**0.7216**	**0.8020**	0.8239	**0.2392**	**0.2141**	**0.2584**	0.2750

- **LASS** [21] is a joint embedding approach that integrates natural-language semantics with structural information of knowledge graphs. It fine-tunes a pre-trained language model using a probabilistic structured loss, aligning semantic representations with relational structures. This joint optimization enhances robustness and achieves strong link prediction performance even in low-resource settings.
- **InstructGLM** [32] is an instruction-tuned language model designed to perform knowledge graph tasks via natural-language instructions. Compared with KGBERT, it demonstrates stronger semantic generalization and better task alignment across unseen schemas.
- **KG-LLM** [13] is a large language model framework that transforms graph structures into textual prompts for multi-hop relational reasoning. It leverages prompt engineering or lightweight fine-tuning, enabling adaptation to new knowledge graphs without extensive retraining.
- **StructGPT** [8] is a structure-aware framework that uses an iterative reading-then-reasoning paradigm: it first employs specialized interfaces to collect relevant structured evidence from graphs (or tabular/DB data), then lets the LLM perform reasoning over both structural priors and semantic context. This design helps improve interpretability and generalization when processing structured data.
- **IO Prompt**[3] is a prompt-based large language model baseline, where the input (entity pair and context) and output (relation label) are formatted in a direct input–output mapping style. It evaluates the model's ability to align structured relational patterns with concise prompt formulations without explicit reasoning chains.
- **CoT Prompt** [35] (Chain-of-Thought Prompting) extends the IO Prompt paradigm by encouraging the model to generate intermediate reasoning steps before producing the final relation prediction. This approach enhances interpretability and supports multi-hop or compositional relational reasoning through explicit step-by-step inference.

For evaluation, we employ **Mean Reciprocal Rank (MRR)** and **Hits@k** ($k = 1, 3, 10$) as metrics. MRR measures the average ranking quality across all candidate relations, while Hits@k evaluates whether the correct relation appears within the top k predictions. To ensure a fair comparison across paradigms, we adopt slightly different candidate sampling strategies: for **KGE** and **PLM**-based methods, each positive triple is evaluated against **24 randomly sampled negative relations**; for **LLM**-based methods, the model directly ranks among **25 candidate relations** provided in the prompt. All experiments follow the same evaluation scripts and data splits to ensure comparability and statistical consistency.

Our implementation is based on the **PyTorch 2.8.0** and **Transformers 4.28+** frameworks. We use the **AdamW** optimizer with an initial learning rate of 3×10^{-4}, linearly warmed up for the first 100 steps and kept constant thereafter. The per-device batch size is 4, resulting in an effective batch size of 16 across two GPUs. Parameter-efficient tuning is performed using **LoRA** with rank $r = 32$, scaling factor $\alpha = 64$, and dropout rate 0.05, applied to all Transformer attention projection matrices (Q, K, V, O). No weight decay is applied, and gradients are clipped to a maximum norm of 1.0. Mixed-precision training (BF16/FP16) is enabled to improve computational efficiency. The main experiments are trained for 3 epochs (1 epoch for FB15k-237, and 3 epochs for CoDEx-S and

DBpedia50, respectively). Validation and checkpoint saving are conducted every 2000 steps, and early stopping is triggered when the validation loss does not improve for three consecutive evaluations. The best-performing checkpoint, measured by validation loss, is used for final testing. All experiments are executed on two NVIDIA RTX 4090 GPUs (48GB × 2).

5.2 Main Results

Table 3 presents the overall performance of KELLM and all baselines across three benchmark datasets. Overall, our model achieves either the best or second-best results in all metrics, demonstrating its strong ability to integrate structural embeddings with semantic reasoning. Notably, KELLM attains the highest performance on CoDEx-S, with an MRR of 0.7627 and Hits@1/3/10 of 0.7216, 0.8020, and 0.8239, respectively. Compared with the second-best model, it achieves relative improvements exceeding 8% in MRR and 7% in Hits@1, confirming its superiority in fine-grained semantic discrimination and complex relational reasoning.

5.2.1 KGE Methods. Traditional knowledge graph embedding methods (TransE, DistMult, ComplEx, and RotatE) exhibit stable performance on structurally dominated datasets such as FB15k-237. These models capture low-order relational regularities and compositional patterns effectively. However, they fail to model higher-order semantics or context-dependent relations, resulting in significant performance drops on semantically richer datasets like CoDEx-S and DBpedia50. Among them, RotatE achieves relatively balanced results due to its geometric interpretability, but its lack of semantic alignment limits generalization.

5.2.2 PLM Methods. PLM-based approaches (KGBERT, BERTRL, and LASS) leverage pre-trained language models to encode textualized triples, enabling stronger semantic representation. They achieve noticeable improvements on CoDEx-S and DBpedia50, demonstrating the benefit of incorporating linguistic priors into relational reasoning. Nevertheless, these models rely heavily on task-specific fine-tuning and often struggle to scale to large, heterogeneous graphs. For example, BERTRL achieves competitive results on FB15k-237 but fails to maintain performance on DBpedia50 due to domain mismatch and sparse contextual coverage.

5.2.3 LLM Methods. LLM-based approaches (KG-LLM, InstructGLM, StructGPT, IO Prompt, and CoT Prompt) represent a new paradigm that unifies language understanding and relational reasoning. Instruction-tuned models such as InstructGLM and StructGPT exhibit improved adaptability and semantic alignment, while prompt-based methods (IO Prompt and CoT Prompt) evaluate the zero/few-shot reasoning capabilities of large models under different prompting paradigms. The IO Prompt baseline tests direct input–output mapping performance, whereas CoT Prompt encourages explicit reasoning chains, leading to better interpretability and moderate performance gains on CoDEx-S. However, LLM-based models generally face challenges in computational efficiency and consistency across datasets: despite their flexibility, they are sensitive to prompt design and often underperform when relational structures are complex or sparse.

5.2.4 Our Model (KELLM). In contrast, KELLM demonstrates consistent and substantial improvements across all benchmarks. Its superiority stems from the deep fusion of structural priors and semantic reasoning: structural embeddings inject geometric constraints, while the LoRA-tuned LLM efficiently captures contextual dependencies and relational semantics. This hybrid design yields balanced reasoning between structure and semantics, achieving robustness across diverse graph schemas. Although KELLM does not obtain the absolute best score on a few metrics (e.g., Hits@10 on FB15k-237 and DBpedia50), the differences remain marginal (within 1–2%), while the overall trend shows remarkable stability and generalization. Particularly on CoDEx-S, KELLM exhibits uniform and significant gains, validating the effectiveness of integrating structured embeddings and semantic reasoning for complex relational prediction.

5.3 Ablation Study

To evaluate the contribution of each component in our framework, we conduct a series of ablation studies under the same settings as the main experiments. The results are reported in Table 4. Specifically, we analyze four model variants: (1) removing structural information (*w/o Structural Info*); (2) removing the Token Translator module (*w/o Token Translator*); (3) removing multi-hop relational information (*w/o Multi-hop Info*); and (4) excluding the supervised fine-tuning stage (*w/o SFT*). Each variant is compared to the full model to quantify the effect of each component.

5.3.1 Effect of Structural Information. Removing all structural information leads to the largest and most consistent performance degradation across datasets, particularly on FB15k-237, where the MRR drops from 0.5871 to 0.5123 (a relative decrease of about 13%). This result highlights the indispensable role of structural priors in guiding relational reasoning and constraining candidate selection. Without explicit graph structure, the model struggles to capture relation topology and compositional regularities, resulting in less accurate predictions.

5.3.2 Effect of Token Translator. Eliminating the Token Translator also results in noticeable performance reduction, especially on CoDEx-S, where MRR declines from 0.7627 to 0.6920. Although the model still has access to structural inputs, the lack of translation between symbolic features and language representations introduces a modality gap, weakening the integration of structural and semantic information. This indicates that the Token Translator is essential for aligning heterogeneous embedding spaces and ensuring effective multimodal fusion.

5.3.3 Effect of Multi-hop Information. When multi-hop relational paths are removed, the model's performance drops moderately, with the most significant decrease observed on CoDEx-S (from 0.7627 to 0.7312 in MRR). This suggests that multi-hop reasoning provides additional contextual cues for capturing long-range dependencies between entities. The impact is smaller on FB15k-237 and DBpedia50, as they contain fewer complex relational chains. Nevertheless, the results confirm that multi-hop semantics improve the model's reasoning depth and interpretability.

Table 4: Ablation study results (MRR and Hits@k).

Model Variant	FB15k-237				CoDEx-S				DBpedia50			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
w/o Structural Info	0.5123	0.4631	0.5412	0.5678	0.6765	0.6279	0.7142	0.7483	0.2083	0.1764	0.2260	0.2457
w/o Token Translator	0.5284	0.4860	0.5541	0.5789	0.6920	0.6443	0.7306	0.7615	0.2135	0.1821	0.2324	0.2538
w/o Multi-hop Info	0.5570	0.5221	0.5843	0.6071	0.7312	0.6900	0.7735	0.7984	0.2281	0.2029	0.2473	0.2661
w/o SFT	0.4692	0.4215	0.4958	0.5237	0.6104	0.5610	0.6513	0.6891	0.1914	0.1622	0.2103	0.2295
Full Model	**0.5871**	**0.5550**	**0.6170**	**0.6330**	**0.7627**	**0.7216**	**0.8020**	**0.8239**	**0.2392**	**0.2141**	**0.2584**	**0.2750**

5.3.4 Effect of Supervised Fine-tuning (SFT). Removing the supervised fine-tuning stage yields the most severe degradation overall, especially on FB15k-237 and DBpedia50, where MRR declines to 0.4692 and 0.1914, respectively. Without SFT, the model cannot effectively adapt the pretrained language model's general semantic knowledge to the structured reasoning objective, resulting in unstable rankings and reduced accuracy. These findings demonstrate that SFT plays a critical role in bridging the gap between semantic understanding and relational prediction.

Across all datasets, the full model consistently outperforms every ablated variant, verifying that each module contributes to the final performance. Structural encoding, token-level alignment, multi-hop reasoning, and supervised fine-tuning jointly form the foundation of our framework. Their synergistic interaction enables effective integration of semantic and structural knowledge, resulting in robust, interpretable, and generalizable relational reasoning.

5.4 Analysis Experiments

To gain a deeper understanding of the factors affecting model performance, we conduct three sets of analytical experiments focusing on: (1) the number of candidate relations, (2) the backbone language model, and (3) the knowledge graph embedding (KGE) method. All experiments are performed under identical training and inference configurations on **FB15k-237**, **CoDEx-S**, and **DBpedia50**. The results are illustrated in Figure 4, and quantitative coverage statistics are reported in Table 5.

5.4.1 Effect of Candidate Number. The number of candidate relations determines the search space during reasoning and directly affects task difficulty and computational complexity. As shown in Table 5, when the number of candidates increases from 20 to 40, the overall model performance slightly decreases across all datasets, with the most significant degradation observed on CoDEx-S, which contains diverse and fine-grained relation types. This pattern reflects the trade-off between search space and ranking precision: smaller candidate sets yield artificially higher scores due to fewer distractors, while larger sets introduce more semantically similar relations that challenge the model's discrimination ability. Statistical analysis further confirms that the proportion of gold relations falling outside the top-k steadily increases with candidate size (e.g., from 87.8% to 98.7% on CoDEx-S), suggesting that ranking difficulty—rather than semantic misunderstanding—is the primary source of performance degradation. Empirically, setting the candidate number to 25 achieves the best balance between fairness, ranking difficulty, and computational cost.

Table 5: Statistics of top-k coverage under different candidate sizes. Each entry reports the total number of test samples, the number of samples where the gold relation appears within top-k, and the corresponding proportion (%).

Dataset	Candidates	Total	Top-k	Proportion (%)
FB15k-237	20	1,9496	16,738	85.9
	25	19,496	16,968	87.0
	30	19,496	17,159	88.0
	35	19,496	17,333	88.9
	40	19,496	17,494	90.0
CoDEx-S	20	1,828	1,605	87.8
	25	1,828	1,648	90.2
	30	1,828	1,696	92.8
	35	1,828	1,748	95.6
	40	1,828	1,804	98.7
DBpedia50	20	10,969	3,538	32.25
	25	10,969	3,619	33.0
	30	10,969	3,675	33.5
	35	10,969	3,738	34.1
	40	10,969	3,789	34.5

5.4.2 Effect of Backbone Model. The scale and architecture of the backbone language model play a decisive role in determining both semantic expressiveness and structural reasoning ability. As the model expands from Qwen2.5-1.5B to Qwen2.5-7B, MRR and Hits@1 consistently improve, but the gain becomes marginal beyond 3B parameters, demonstrating the diminishing return of scaling. At comparable scales, Qwen-based backbones consistently outperform LLaMA counterparts, reflecting the advantage of instruction-tuned objectives and fact-preserving pretraining for structured reasoning. It is important to note that all experiments are conducted entirely in English, indicating that the improvement stems from architectural and optimization factors rather than linguistic bias. Overall, Qwen2.5-3B provides the best trade-off between reasoning accuracy, efficiency, and stability.

5.4.3 Effect of Knowledge Graph Embedding (KGE) Method. Different KGE methods provide distinct structural priors to guide reasoning. TransE captures simple one-to-one and hierarchical relations effectively but struggles with symmetric or compositional patterns. DistMult and ComplEx mitigate symmetry issues via bilinear and complex-space interactions, yet remain insufficient for modeling long-range dependencies. In contrast, RotatE consistently

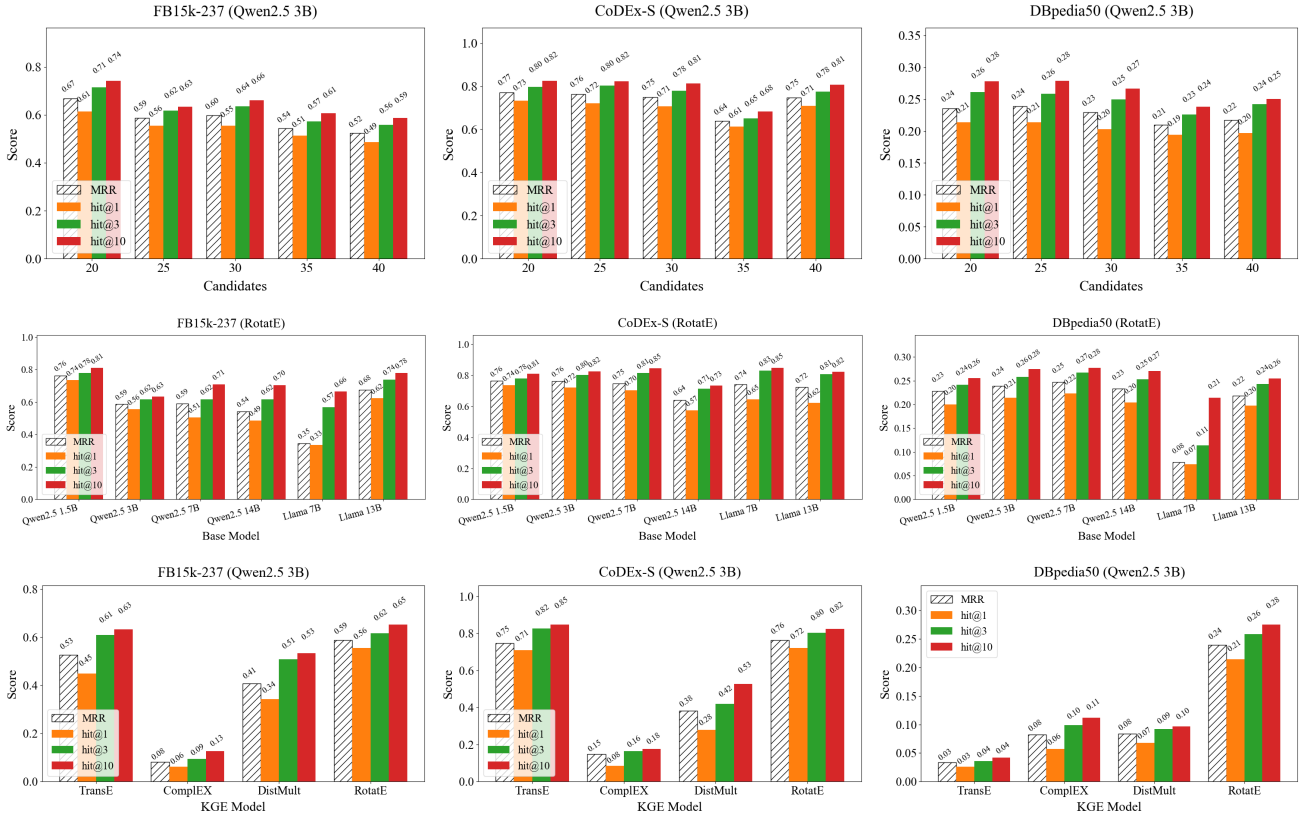

Figure 4: Comprehensive analysis of key factors influencing model performance. Top row: effect of candidate number; middle row: backbone model comparison; bottom row: KGE method comparison. Each column corresponds to one dataset (left: FB15k-237, middle: CoDEx-S, right: DBpedia50). All metrics include MRR and Hits@k, revealing complementary performance trends across datasets.

achieves the best or near-best performance across all datasets, owing to its geometric interpretability and capacity to model diverse relation types. Furthermore, its rotational representation aligns more naturally with the LLM's embedding geometry, facilitating smooth fusion under LoRA-based fine-tuning. This synergy between symbolic structure and contextual semantics enhances both interpretability and robustness across datasets.

Collectively, these findings reveal that the candidate number controls the reasoning scope, the backbone model determines semantic expressiveness, and the KGE method governs structural grounding. Together, these factors shape the overall trade-off among accuracy, interpretability, and computational efficiency, offering valuable insights for scaling KELLM to larger graphs and more open-domain reasoning scenarios.

6 Conclusion

This paper presents a unified framework, **KELLM**, which effectively bridges structured knowledge modeling and semantic reasoning. The framework leverages knowledge graph embedding models for candidate relation filtering to reduce computational complexity in large-scale graph reasoning. Through the proposed *Token Translator* module, continuous embeddings are mapped into symbolic representations compatible with large language models, while multi-hop path information is incorporated to enhance semantic modeling and interpretability. Experimental results on multiple public benchmarks demonstrate that KELLM consistently outperforms embedding-based, PLM-based, and LLM-based methods in terms of MRR and Hits@k, confirming the effectiveness and scalability of integrating structural and semantic knowledge for relation prediction.

Despite its promising results, KELLM still faces several limitations. Its reasoning efficiency and knowledge coverage remain constrained when applied to extremely large-scale or multilingual knowledge graphs. The mapping mechanism of the Token Translator is currently static and lacks adaptability to task- or domain-specific variations. Moreover, this work mainly focuses on relation prediction, leaving the extension to more complex downstream tasks such as entity prediction, fact verification, and interpretable reasoning for future exploration. Future work will aim to further improve inference efficiency and system deployability, investigate dynamic structure–semantic alignment mechanisms, and extend the framework to open-domain reasoning, multimodal knowledge

graph completion, and intelligent question answering, thereby advancing the integration of database systems and artificial intelligence.

Artifacts

All artifacts related to this paper are publicly available at the project repository: https://github.com/TheBlueBanisters/KELLM. The repository contains the implementation of the KELLM framework, pretrained models, datasets, and experiment scripts necessary to reproduce the results reported in this paper.

References

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, Proceedings of the 6th International Semantic Web Conference (ISWC)*. 722–735.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*. 2787–2795.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS) 2020 (NeurIPS '20, Vol. 33)*. Curran Associates, Inc., Red Hook, NY, 1877–1901. doi:10.5555/3495724.3495883

[4] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations (ICLR)*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.

[6] Gonzalo I. Diaz, Achille Fokoue, and Mohammad Sadoghi. 2018. EmbedS: Scalable, Ontology-aware Graph Embeddings. In *Proceedings of the 21st International Conference on Extending Database Technology (EDBT) (EDBT '18)*. 433–436. doi:10.5441/002/edbt.2018.40

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

[8] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Singapore, 9237–9251. doi:10.18653/v1/2023.emnlp-main.574

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations (ICLR)*.

[10] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3045–3059.

[11] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K"uttler, Mike Lewis, Wen tau Yih, Tim Rockt"aschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP. In *Advances in Neural Information Processing Systems (NeurIPS)*. 9459–9474.

[12] Jinpeng Li, Hang Yu, Xiangfeng Luo, and Qian Liu. 2024. COSIGN: Contextual Facts Guided Generation for Knowledge Graph Completion. In *NAACL-HLT*. 1669–1682.

[13] Qian Li, Zhuo Chen, Cheng Ji, Shiqi Jiang, and Jianxin Li. 2024. LLM-based Multi-Level Knowledge Generation for Few-shot Knowledge Graph Completion. In *IJCAI*. 2135–2143.

[14] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*. 4582–4597.

[15] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *ACL*. 1906–1919.

[16] Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 43–54. doi:10.18653/v1/D19-1005

[17] Fabio Petroni, Tim Rockt"aschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2463–2473.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.

[19] Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *EMNLP*. 8328–8350.

[20] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-Sequence Knowledge Graph Completion and Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2814–2828.

[21] Jianhao Shen, Chenguang Wang, Linyuan Gong, and Dawn Song. 2022. LASS: Joint Language Semantic and Structure Embedding for Knowledge Graph Completion. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*. 1976–1988.

[22] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized Language and Knowledge Embedding. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. 3660–3670.

[23] Yizhou Sun, Brian Norick, Jiawei Han, Xifeng Yan, Philip S Yu, and Tian Yu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proceedings of the VLDB Endowment (PVLDB)* 4, 11 (2011), 992–1003.

[24] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.

[25] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA Model. https://github.com/tatsu-lab/stanford_alpaca. GitHub repository.

[26] Komal K. Teru, Etienne Denis, and William L. Hamilton. 2020. Inductive Relation Prediction by Subgraph Reasoning. In *ICML*. 9448–9457.

[27] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing Text for Joint Embedding of Text and Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1499–1509.

[28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *International Conference on Machine Learning (ICML)*. 2071–2080.

[29] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations (ICLR 2020)*. OpenReview, Addis Ababa, Ethiopia. https://openreview.net/forum?id=BylA_C4tPr

[30] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. In *Proceedings of The Web Conference 2021 (WWW '21)*. 1737–1748. doi:10.1145/3442381.3450043

[31] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 172–181.

[32] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. 2024. InstructGraph: Boosting Large Language Models via Graph-centric Instruction Tuning and Preference Alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*. 13492–13510. doi:10.18653/v1/2024.findings-acl.801

[33] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. 4281–4294.

[34] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *TACL* 9 (2021), 176–194.

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS) 2022 (NeurIPS '22, Vol. 35)*. Curran Associates, Inc., Red Hook, NY. doi:10.5555/3600270.3602070

[36] Ruqing Xie, Zhiyuan Liu, Jia Jia, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *AAAI*. 2659–2665.

[37] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations (ICLR)*.

[38] Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *NeurIPS*. 2319–2328.

[39] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. *arXiv preprint arXiv:1909.03193* (2019).

[40] Hanwen Zha, Zhiyu Chen, and Xifeng Yan. 2022. Inductive Relation Prediction by BERT. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022)*. AAAI Press, 5923–5931. doi:10.1609/aaai.v36i5.20537