LGC-CR: Few-shot Knowledge Graph Completion via Local Global Contrastive Learning and LLM-Guided Refinement

Yiming Xu
xym30@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Qi Song*
qisong09@ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Yihan Wang
yihanw@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Wangqiu Zhou
rafazwq@mail.ustc.edu.cn
Hefei University of Technology
Hefei, Anhui, China

Junli Liang
jlliang@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Abstract

Recent years have witnessed increasing interest in few-shot knowledge graph completion (FKGC), which aims to infer novel query triples for few-shot relations from limited references. Despite promising progress, existing methods face two key challenges: (1) They often overlook rich higher-order neighbors, while traditional high-order aggregation methods are prone to introducing noise and lack effective alignment across multi-view neighborhood information. (2) Meta-learning methods over-rely on embeddings, making them susceptible to spurious relational patterns. Meanwhile, LLM-based methods, despite their potential, suffer from hallucinations and input constraints. To this end, we propose a novel framework that combines meta-learning, enhanced via a **L**ocal-**G**lobal **C**ontrastive network, with LLM-guided **C**ontextual **R**efinement (LGC-CR). At the data level, we design a local-global contrastive network to jointly aggregate relevant local features and capture stable global representations while filtering high-order noise, then align these two views through a dual contrast module to ensure consistency. At the model level, we employ an LLM refinement module, which retrieves relevant contexts to construct prompts and applies a knowledge selector to identify high-quality facts based on diversity and centrality, enabling efficient fine-tuning of LLMs to refine the preliminary predictions of meta-learning. The experimental results demonstrate that LGC-CR delivers better and more robust performance than state-of-the-art baselines, with Hit@1 improvements of 8.1%, 21.7%, and 20.6% on NELL, Wiki, and FB15K, respectively.

CCS Concepts

• **Computing methodologies → Machine learning**.

Keywords

Few-shot Knowledge Graph Completion; Contrastive Learning; Large Language Model; Graph Neural Network

1 Introduction

Knowledge Graphs (KGs), serving as structured semantic knowledge bases [35, 37], consist of a series of factual triples in the format of (*head entity*, *relation*, *tail entity*). KGs play a pivotal role in intelligent applications, including question-answering [5, 31], recommendation systems [9], and retrieval-augmented generation [13, 29]. However, with the continuous emergence of new knowledge, real-world KGs often face the challenge of incompleteness [6], manifested as missing or underrepresented relations and entities. This motivates research on Knowledge Graph Completion (KGC), which aims to infer missing elements in incomplete facts. Typical KGC methods address this task by learning embeddings based on either graph structures [2, 41] or text descriptions [25, 28].

Despite their success, embedding-based KGC methods often require abundant training triples for a certain type of relation. However, relations in real-world KGs usually exhibit a long-tail distribution, with most types of relations being associated with only a few triples. About 10% of the relation types in the Wikidata dataset contain fewer than 10 triples [3]. Moreover, in practical scenarios [10] where KGs are dynamically updated, newly introduced relations

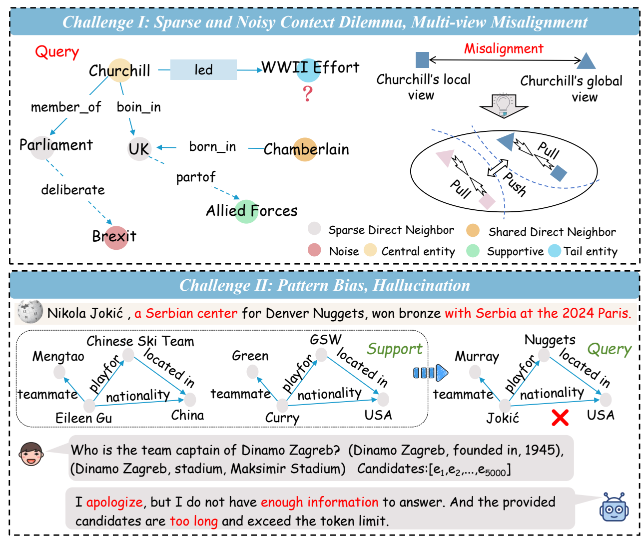


Figure 1: Two challenges in FKGC. (Top) Noisy, underutilized high-order neighbors and misaligned local/global views. (Bottom) Pattern bias in meta-learning and hallucination in LLM.

*Corresponding author.

often face the cold-start problem with no available triples, which leads to a sharp performance drop for these methods.

To address this challenge, few-shot knowledge graph completion (FKGC) methods have been proposed, most of which aim to predict the missing tail entity t in a query triple $(h, r, ?)$ only given K support triples [39] for the few-shot relation r, where K is typically a small number. Current methods include model-agnostic meta-learning methods (e.g., MetaR [3], GANA [27], CFKGC [22]), which aim to learn the features of associated triples in training tasks for generalization to new tasks, and metric-based methods (e.g., FAAN [30], CIAN [19], SuperRL [11]), which learn matching functions from training tasks and apply them to unseen tasks. Since task relation embeddings are unknown, previous methods mostly leverage the local neighbors of entity pairs to enhance relation prototype learning. Moreover, with the rapid development of large language models (LLMs), some recent methods [1, 16, 18, 43] directly use LLMs for graph completion, which aim to exploit their pretrained knowledge and contextual reasoning capabilities. Despite progress, current approaches still face two major challenges:

Challenge 1: Existing methods struggle to leverage noisy high-order neighborhoods and align multi-view information. They over-rely on local neighbors while ignoring the rich semantics of high-order neighborhoods. Traditional aggregation methods often introduce noise and struggle to integrate multi-view neighborhood information. As shown in Figure 1, for query (*Churchill*, *led*, *WWII Effort*), the direct neighbors(such as *UK* and *Parliament*) only offer superficial details related to nationality or employment, failing to reflect Churchill's leadership role in *World War II*. Furthermore, many entities(e.g., *Churchill* and *Chamberlain*) in KGs share common local neighbors, making it difficult to distinguish their roles. In contrast, high-order neighbors like *Allied Forces* in the global view enrich *Churchill*'s portrayal in *World War II*, but may also introduce irrelevant noise (e.g., (*Parliament*, *deliberate*, *Brexit*)). Additionally, representations from local and global views may be misaligned, which undermines reasoning stability and robustness.

Challenge 2: Embedding-based meta-learners are vulnerable to pattern bias, while LLMs suffer from hallucination and input constraints. As shown in Figure 1, embedding-based meta learning may overfit to frequent relational patterns in the support set (e.g., *Eileen Gu* and *Curry*), which can mislead inference on query entities like *Nikola Jokić*, especially for long-tail cases. On the other hand, LLM-based methods, while powerful in language understanding, are susceptible to hallucination due to a lack of contextual support and suffer from input length limitations that hinder the processing of complex queries or large candidate sets.

To address the above challenges, we propose a novel FKGC framework (LGC-CR) that augments both the data and model levels by integrating enhanced meta-learning with LLM-guided refinement. **At the data level, to address Challenge 1, we design a local-global contrastive network to enrich and align neighborhood information**: local features are aggregated via cross-attention, while global representations are captured through a hierarchical module composed of Residual Graph Convolutional Network (ResGCN) and Filter Graph Attention (FilGAT) layers, where ResGCN ensures stable first-order aggregation through residual connections, and FilGAT leverages structure-semantic entity dissimilarity to dynamically filter irrelevant high-order neighbors. These two views are then aligned through dual contrastive learning, promoting consistency between local and global perspectives. To further enhance relational discrimination, we additionally introduce a fusion similarity network that constructs a learnable non-linear metric space. **At the model level, to address Challenge 2, we introduce an LLM refinement**, which retrieves relevant structural and semantic contexts to construct prompts, and incorporates a knowledge selector that identifies high-quality facts using a dual criterion of diversity and centrality, enabling efficient fine-tuning of LLMs to optimize the preliminary results of meta-learning.

In summary, our contributions are as follows:

- We propose a novel FKGC framework, named LGC-CR, which combines an enhanced meta-learning model with LLM knowledge, effectively addressing challenges in high-order neighborhood noise, multi-view information misalignment, and mitigating both spurious relational patterns and hallucinations.
- To effectively utilize high-order neighbors and align multi-view information, we propose a local-global contrastive network. Relevant local neighborhood information is aggregated via cross-attention, while stable and denoised global information is captured by a ResGCN-FilGAT hierarchical graph network. To enhance the consistency between these two views, dual contrastive learning is introduced. Moreover, a fusion similarity network is designed to construct a learnable non-linear metric space.
- To reduce spurious pattern interference in meta-learning and hallucinations in LLMs, we propose an LLM-guided refinement method, which retrieves structural and semantic contexts relevant to the query and introduces a knowledge selector to filter high-quality facts based on diversity and centrality, fine-tuning the LLM to optimize the initial results of the meta-learning model.
- Experiments on three benchmark datasets demonstrate that LGC-CR outperforms the state-of-the-art baselines by 8.1-20.6% in terms of Hit@1 and shows robust performance in few-shot scenarios. The case study shows that LGC-CR effectively distinguishes between the positive and negative entity pairs.

2 Related Work

Few-shot KG Completion. Current approaches for few-shot KG completion can be classified into two categories: (1) Metric-based methods: GMatching [39] introduces a local neighbor encoder to enhance entity embeddings but assumes equal contribution from all adjacent relations. FAAN [30] proposes a one-hop neighbor encoder with adaptive attention and a Transformer encoder to capture dynamic interactions between entities and relations. CIAN [19] further models the interaction between head and tail entities using an attention mechanism. SuperRL [11] selects crucial neighbor entities for few-shot relations from triplet and context perspectives. (2) Optimization-based methods: MetaR [3] designs a fast gradient descent update to transfer relational meta-information from support triples to queries. GANA [27] introduces a gated attention-based neighbor aggregator to filter noise in 1-hop neighbors.

Despite existing efforts, two issues remain: (1) they fail to leverage rich yet noisy high-order neighborhood information, and ignore the alignment of multi-view neighborhoods; (2) meta-learning methods over-rely on embeddings, making models susceptible to spurious relational patterns learned from support sets.

Contrastive Learning on Graph. The core of contrastive learning is to create diverse input views to enhance the model's generalization and robustness. Existing methods either create multiple views using features like node attributes and relation types [24, 33], or generate views by perturbing the data [34, 38]. However, these methods are not designed for knowledge graphs (KGs) and cannot be directly applied to few-shot knowledge graph completion.

LLM-Enhanced KG Completion. Recently, large language models (LLMs) have attracted widespread attention and been used for knowledge graph completion. RPLLM [14] fine-tunes Llama2-7B using node names from the knowledge graph. GenKGC [16] prompts LLMs to transform triples into context-rich segments. However direct use of LLMs for completion struggles with understanding knowledge graph structure and semantics, resulting in poor performance and hallucinations. On the other hand, LLMs face input length limitations, preventing them from handling large candidate entities. KIC-GPT [36] combines LLMs with traditional KGC methods, leveraging both KG and LLM knowledge. However, accessing closed-source LLMs like ChatGPT is costly, and it is not designed for few-shot KG completion.

3 Design of LGC-CR Framework

3.1 Notations and Definitions

First we give a summary of primary notations in Table 1.

Knowledge Graph (KG) is represented as a set of triples $\mathcal{G} = \{(h, r, t) \in \mathcal{E}\times\mathcal{R}\times\mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ are entity set and relation set, $r \in \mathcal{R}$, while $h, t \in \mathcal{E}$ denote the head and tail entity respectively.

Few-shot knowledge graph completion (FKGC) is a specialized task proposed for the relations with only a few triples, which are called few-shot relations. Each few-shot relation r corresponds to one knowledge graph completion task T_r. Given a task relation r and its support set $S_r = \{(h_i, r, t_i)\}_{i=1}^{K}$, the objective is to predict the missing tail entities for the query triples, where the few-shot size K is typically very small. Upon sufficient training on $\mathcal{R}_{\text{train}}$, the trained model can be leveraged for the meta-validation and meta-testing phases using $\mathcal{R}_{\text{valid}}$ and $\mathcal{R}_{\text{test}}$, respectively.

3.2 Framework Overview

As shown in Figure 2, the overall framework of LGC-CR is composed of three components: local-global contrastive network, fusion-based similarity network, and LLM-guided refinement. Local-global contrastive network adopts a cross-attention mechanism to capture relevant local neighborhood information and employs a ResGCN-FilGAT hierarchical network to extract stable and denoised global features. These two enhanced views are then aligned through dual contrastive learning. Building on the enriched and aligned entity representations, fusion-based similarity network enhances the model's ability to compare queries with relation prototypes by integrating three complementary similarity metrics. To further distinguish positive samples from negatives, a hard negative reasoning loss is introduced and jointly optimized with the contrastive loss. Finally, LLM-guided refinement retrieves relevant contexts and employs a knowledge selector to identify diverse and central facts, which are then used to efficiently fine-tune the LLM and refine the meta-learning results.

Table 1: Notations used in this paper.

Symbol	Definition
$\mathcal{G}$	knowledge graph
T_r	few-shot task corresponding to relation r
S_r, Q_r	support and query set corresponding to relation r
K	the number of shots for FKGC
N	the semantic-aware neighbor set
local neighbor	1-hop neighbors directly connected to the entity
global neighbor	multi-hop neighbors (1-hop and high-order)
$\mathbf{h}_t, \mathbf{t}_t$	the task-aware representations
$\mathbf{h}_s, \mathbf{t}_s$	local entity representations
D_{ij}	structure-semantic entity dissimilarity
$\mathbf{h}_g, \mathbf{t}_g$	global entity representations
E_q	adaptive relational prototype
$CN_{(p_q, E_q)}$	learnable compare network metric
$\mathcal{K}$	number of relation clusters in diversity sampling

3.3 Local-Global Contrastive Network

The module consists of three parts: (1) Cross-Attention Local Enhancement, (2) ResGCN-FilGAT Global Enhancement, and (3) Dual-View Contrastive Learning.

3.3.1 Cross-Attention Local Enhancement. To obtain more complete neighbor information while maintaining semantic relevance, we propose a Knowledge-Enhanced Neighbor Sampling (KNS), which ranks the neighbor set based on textual description. This approach overcomes the incompleteness issues of ID-based sampling [17] and the noise problems of random sampling methods [32]. Specifically, before training, we collect all first-hop neighbors of each entity to construct a complete neighbor set N'. Next, we concatenate the description information of each neighbor relation with the corresponding entity's description, and encode them using a pre-trained BERT [4] model. By computing their similarity to the center entity's embedding, we rank the neighbors and select the most relevant ones to form a semantic-aware neighbor set N(e.g., $N_h = \{(r_{hi}, e_{hi})\}$). This step is performed only once before training. After constructing the neighbor set, we use a cross-attention mechanism [19] to aggregate local neighbors, where the cross-attention mechanism includes task-aware attention and entity-pair-aware attention. The task-aware attention sub-module processes an entity pair (h, t) regarding to the task relation. It takes the local neighbors of each entity as input and outputs the corresponding task-aware entity representation as follows:

$$\bar{\mathbf{h}}_t = W_t^s \left(\text{softmax}(Q_t K_t^T) V_t\right), \tag{1}$$

where

$$Q_t = W_t^Q \left(W_t(\mathbf{h} \oplus \mathbf{t}) + \mathbf{b}_t\right), \; K_t = W_t^K \mathbf{r}_{hi}, \; V_t = W_t^V \mathbf{n}_{hi}, \tag{2}$$

$$\mathbf{n}_{hi} = \text{ReLU}\left([\mathbf{r}_{hi} \oplus \mathbf{e}_{hi}]\, W^N + b_n\right). \tag{3}$$

Here, $\mathbf{h}, \mathbf{t}, \mathbf{r}_{hi}, \mathbf{e}_{hi} \in \mathbb{R}^d$ are the embedding vectors for h, t, r_{hi}, e_{hi}, respectively. The operation $\oplus$ denotes concatenation. $W^N \in \mathbb{R}^{2d\times d}$, $b_n \in \mathbb{R}^d$ are learnable parameters for all neighbors. $W_t^Q, W_t^K, W_t^V, W_t^s \in \mathbb{R}^{d\times d}, W_t \in \mathbb{R}^{2d\times d}, b_t \in \mathbb{R}^d$ are all trainable parameters.

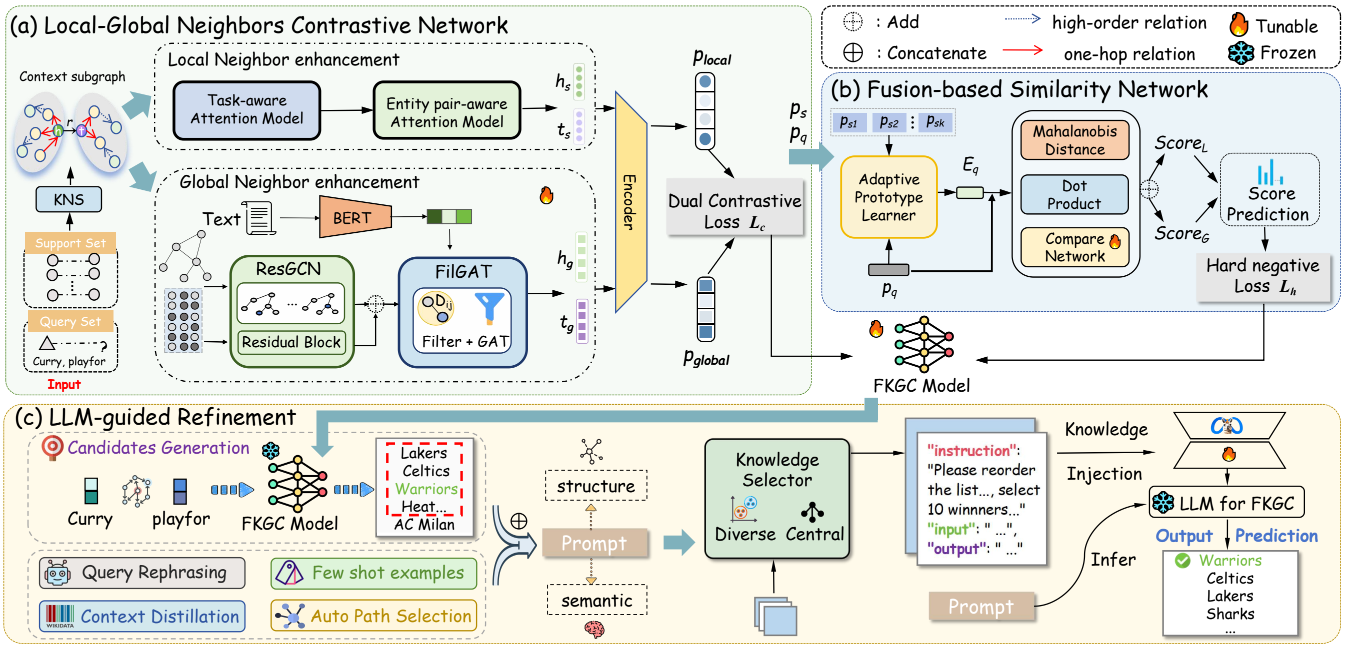


Figure 2: Overview of our framework LGC-CR.

To emphasize the entity itself, we perform a residual-like computation to get the task-aware representation of the entity:

$$\mathbf{h}_t = \text{ReLU}\left(\mathbf{h}W_{\text{ent}} + \bar{\mathbf{h}}_t W_{\text{nbr}}\right), \tag{4}$$

where $W_{\text{ent}}, W_{\text{nbr}} \in \mathbb{R}^{d\times d}$ are two trainable weight matrices. Similarly, we apply the same method to the tail entity t and its neighbors N_t, resulting in the task-specific representation $\mathbf{t}_t$.

The entity-pair-aware attention sub-module determines the semantic relevance between the head and tail entities by allowing each entity to focus on the neighbors of its paired entity. It is implemented in the same way as the task-aware module, where the task-aware representations $\mathbf{h}_t$ and $\mathbf{t}_t$ are input to obtain the final local entity representations $\mathbf{h}_s$ and $\mathbf{t}_s$.

3.3.2 ResGCN-FilGAT Global Enhancement. To capture rich global neighborhood information while avoiding noise from higher-order neighbors, we propose a ResGCN-FilGAT hierarchical graph neural network. ResGCN aggregates low-order neighbors, while FilGAT selectively identifies and incorporates important high-order neighbors. This dual-component approach enables our model to capture more stable and comprehensive neighborhood information while effectively filtering out noise.

Specifically, we design a ResGCN to aggregate the one-hop neighborhood information. First, we aggregate local neighbors using a GCN. To enhance training stability and mitigate gradient vanishing in deep GCNs, we introduce a residual block that directly links the original input to the output. Given the embedding matrix $X \in \mathbb{R}^{n\times d}$ and the adjacency matrix $A \in \mathbb{R}^{n\times n}$ of all entities, where X comes from pretraining embedding provided by TransE [2], n represents the number of entities and d is the embedding dimension. Then we input X and A into the ResGCN as follows:

$$H^{(1)} = \text{ReLU}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW^{(0)}\right) + W^r X, \tag{5}$$

where $H^{(1)}$ represents the output obtained from the ResGCN layer, $\tilde{A}$ is the adjacency matrix with self-loops. The matrix $\tilde{D}$ represents degree matrix of $\tilde{A}$, $W^{(0)} \in \mathbb{R}^{d\times d}$ denotes the weight matrix that can be learned, while W^r denotes residual projection weight.

To extract higher-order semantic information and suppress interference from noisy neighbors, we introduce FilGAT in the second layer. Unlike the traditional Graph Attention Network (GAT) that performs weighted aggregation of all neighbor node features, the core of FilGAT is "not all neighbors are worth considering". The method incorporates an adaptive neighbor filtering mechanism which, in contrast to rigid top-k strategies, dynamically learns how many neighbors each entity should ignore, allowing the network to fuse only valuable higher-order neighborhood information.

The basic idea of FilGAT is shown in Figure 3. Specifically, we first calculate the initial attention scores between entities.

$$o_{ij} = \frac{\exp(\text{LeakyReLU}(a^\top[W^l h_i^l \oplus W^l h_j^l]))}{\sum_{k\in N_i}\exp(\text{LeakyReLU}(a^\top[W^l h_i^l \oplus W^l h_k^l]))}, \tag{6}$$

where $h_i^l, h_j^l \in H^{(1)}$ represent the entity embeddings obtained from the first layer, $a \in \mathbb{R}^{2d}$ is a learnable vector of attention parameters, $W^l \in \mathbb{R}^{d\times d}$ is a weighted matrix for feature mapping. To dynamically adjust attention range, we introduce entity-entity dissimilarity for determining which neighbors to filter. We enhance the original structural embeddings with semantic information from entity textual descriptions, then use these combined representations to calculate dissimilarity. The formula is as follows:

$$D_{ij} = \left\|[h_i \oplus d_i] - [h_j \oplus d_j]\right\|^2, \tag{7}$$

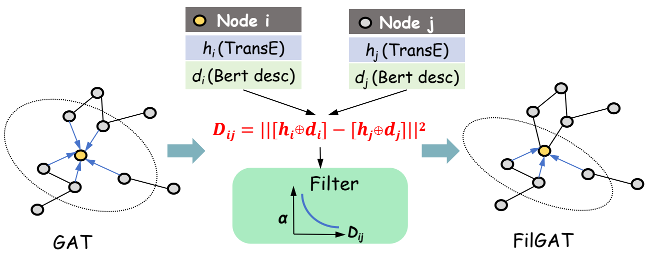


Figure 3: Schematic illustration of FilGAT.

where h_i, h_j are features of entities in a GNN layer, d_i, d_j are the corresponding textual embeddings obtained from the BERT model, and $\| \cdot \|^2$ denotes the Euclidean distance.

Based on the dissimilarity, we propose a neighbor filtering module to refine the attention in Eq.6, yielding the final attention score:

$$\alpha_{ij} = \frac{o_{ij} \cdot \mathrm{sigmoid}(-\beta D_{ij})}{\sum_{k \in N_i} o_{ik} \cdot \mathrm{sigmoid}(-\beta D_{ik})}, \tag{8}$$

where $\beta \in (0, 1]$ is used to control the importance of D_{ij}.

The final attention score decreases as D_{ij} increases. Unrelated neighbors are further weakened and nearly excluded, allowing more attention to be focused on neighbors with higher relevance to the central entity. Based on the final attention scores, FilGAT aggregates the features of entities and their neighbors to generate the output, as shown in the following formula:

$$H_i^{(2)} = \sum_{j \in N(i)} \alpha_{ij} W^l H_j^{(1)}, \tag{9}$$

where $N(i)$ denotes entity i and its neighbors, and we can get the head-tail entity's global representation $\mathbf{h}_g, \mathbf{t}_g \in H^{(2)}$, which capture stable and denoised global neighborhood information.

3.3.3 *Dual-View Contrastive Learning.*

Although local and global entity enhancement effectively capture neighborhood information, representations obtained through different views and methods are often inconsistent, which may introduce noise, especially in few-shot scenarios. To address this issue, we introduce a dual-view contrastive learning module to align the representations enhanced by cross-attention and hierarchical graph neural networks. Specifically, for each query $q \in Q_r$, we enhance the query entity using both cross-attention mechanism and ResGCN-FilGAT hierarchical graph neural networks, obtaining local-level neighbor representations $\mathbf{h}_s/\mathbf{t}_s$ and global-level neighbor representations $\mathbf{h}_g/\mathbf{t}_g$. We then compute the semantic representation of the entity pair by a position-wise feedforward neural network:

$$\begin{aligned} p^q_{local} &= \mathrm{ReLU}([\mathbf{h}_s \oplus \mathbf{t}_s]\, W_{p1} + b_{p1})W_{p2} + p(h,t), \\ p^q_{global} &= \mathrm{ReLU}(\left[\mathbf{h}_g \oplus \mathbf{t}_g\right] W_{p3} + b_{p2})W_{p4} + p(h,t), \end{aligned} \tag{10}$$

where $W_{p1}, W_{p2}, W_{p3}, W_{p4} \in \mathbb{R}^{2d \times 2d}$ and $b_{p1}, b_{p2} \in \mathbb{R}^{2d}$ are trainable matrices, and $p(h,t)$ is the concatenation of h and t. It allows the two query embeddings to be projected into a shared semantic space for contrastive training. Similarly, we generate dual entity pair embeddings $p^{q^-}_{local}$ and $p^{q^-}_{global}$ for each negative query q^-.

If taking the query representation generated by the local encoder as the anchor, the global encoder can be viewed as an augmented encoding process, and vice versa. The local and global representations of the same query form positive pairs $(p^q_{local}, p^q_{global})$, while the local and global representations of positive and negative queries form negative pairs $(p^q_{local}, p^{q^-}_{global})$. Based on this, we define two contrastive losses to maximize the consistency between dual entity-pair embeddings for positive queries and enforce divergence between those of positive and negative queries, as follows:

$$\begin{aligned} L_{local} &= \frac{1}{|Q_r|} \sum_{q \in Q_r} -\log \frac{\exp(\mathrm{sim}(p^q_{local}, p^q_{global})/\tau)}{\sum_i \exp(\mathrm{sim}(p^q_{local}, p^{q_i^-}_{global})/\tau)}, \\ L_{global} &= \frac{1}{|Q_r|} \sum_{q \in Q_r} -\log \frac{\exp(\mathrm{sim}(p^q_{global}, p^q_{local})/\tau)}{\sum_i \exp(\mathrm{sim}(p^q_{global}, p^{q_i^-}_{local})/\tau)}, \end{aligned} \tag{11}$$

where τ is the temperature parameter, and the final dual contrastive loss integrates these two complementary loss terms as:

$$L_c = L_{global} + L_{local}. \tag{12}$$

3.4 Fusion-based Similarity Network

3.4.1 *Prototype Learner and Similarity Network.*

This module comprises an adaptive relational prototype learner and a similarity network combining three complementary metric methods. First, to mitigate noise from the support set during prototype generation, we incorporate query information. Specifically, given a query representation p_q, where $q \in Q_r$, the corresponding prototype E_q for relation r can be computed as:

$$E_q = \sum_{i=1}^{K} \eta_i p_{s_i}, \quad \eta_i = \frac{e^{p_q \odot p_{s_i}}}{\sum_{k=1}^{K} e^{p_q \odot p_{s_k}}}, \tag{13}$$

where p_q denotes embedding of entity pairs from p^q_{global} or p^q_{Local}, η_i denotes attention score between the p_q and p_{s_i}. p_{s_i} represents the i-th support entity pairs, and $\odot$ represents dot product.

Existing methods mostly rely on a single similarity measure, only discriminative in a single feature space, which easily causes similarity bias in few-shot scenarios. To address this, we propose a fusion similarity network that combines dot product, Mahalanobis distance, and compare network. The Mahalanobis distance captures dependencies between features via covariance matrix, while the compare network learns deep, nonlinear metrics among features.

$$dp_{(p_q, E_q)} = p_q \odot E_q, \tag{14}$$

$$Ma_{(p_q, E_q)} = \sqrt{(p_q - E_q)^\top \Sigma^{-1} (p_q - E_q)}, \tag{15}$$

$$CN_{(p_q, E_q)} = \mathrm{ReLU}(Seq_{\Pi}(Seq_I([p_q \oplus E_q])W_1 + b_1)W_2 + b_2), \tag{16}$$

where $Seq(z) = \mathrm{dropout}(\mathrm{LeakyReLU}(\mathrm{LayerNorm}(zW_z + b_z)))$, W_1, W_2, W_z, b_1, b_2, b_z are learnable parameters, Σ^{-1} denotes the inverse of the covariance matrix. *LayerNorm* $(\cdot)$ is a normalization method and *dropout*$(\cdot)$ is used to alleviate overfitting. Then, the three similarity scores are fused as follows:

$$s(p_q, E_q) = dp_{(p_q, E_q)} + \omega \cdot CN_{(p_q, E_q)} - Ma_{(p_q, E_q)}, \tag{17}$$

where ω is a hyperparameter. Since p_q comes from both local and global views, we can obtain the similarity scores $s_G(p_q, E_q)$ and $s_L(p_q, E_q)$ from the global and local perspectives, respectively. Based on this, the final similarity score is computed as:

$$s_f(p_q, E_q) = \theta \cdot s_G(p_q, E_q) + s_L(p_q, E_q). \tag{18}$$

3.4.2 Hard Negative Loss and Optimization. We design a hard negative reasoning loss using the above similarity score. Specifically, for a given relation r, we sample a batch of triples as the positive query set $Q_r^+ = \{(h_q, t_q^+) \mid (h_q, r, t_q^+) \in G_r\}$ and generate the negative query set by randomly polluting the tail entities: $Q_r^- = \{(h_q, t_q^-) \mid (h_q, r, t_q^-) \notin G_r\}$, where G_r denotes background KG. Different from previous works [19, 20], we do not treat all negative samples equally. Instead, we adopt an attention mechanism to focus more on hard negative samples that are difficult to distinguish during training, as shown below:

$$\begin{aligned} \delta_{r_i} &= \frac{\exp(s_f(r_i^q, S_r))}{\sum_{j=1}^{J} \exp(s_f(r_j^q, S_r))}, \\ L_h &= \sum_r \sum_{r^{q+} \in Q_r^+} \left[\gamma - s_f(r^{q+}, S_r) + \sum_{j=1}^{J} \delta_{r_i} s_f(r_i^q, S_r) \right]_+, \end{aligned} \tag{19}$$

where J denotes the number of negative samples, γ is a margin hyperparameter, and $[x]_+ = \max(0, x)$ is the standard hinge loss.

The model is jointly optimized by the Hard Negative Loss and the Dual Contrastive Loss. The overall objective is defined as:

$$L = L_h + \lambda L_c, \tag{20}$$

where λ controls the influence of the contrastive loss.

3.5 LLM-guided Contextual Refinement

To address challenge 2 mentioned above, we propose an LLM-guided Contextual Refinement module. This module retrieves structural and semantic information, and selects high-quality knowledge based on diversity and centrality to fine-tune a dedicated LLM for FKGC, refining the meta-learning-derived initial predictions. It consists of three components: (1) Instruction Retrieval, (2) Knowledge Selector, and (3) Re-ranker via Instruction Fine-tuning.

3.5.1 Instruction Retrieval. For a given query $(h, r, ?)$, we retrieve relevant information to construct the prompt. This retrieval process consists of five key components: **Query Rephrasing**, **Few-shot Examples**, **Context Distillation**, **Auto Path Selection**, and **Candidates Generation**. We illustrate this process using (*Stephen Curry*, *playfor*, ?) as an illustrative example.

<u>Query Rephrasing</u> transforms the structured query (Curry, playfor, ?) into multiple natural language questions using an LLM, such as "Which team does Curry play for?" and "What team is Curry currently on?", and feeds them collectively as prompts to help the model better capture query intent and improve generalization.

<u>Few-shot Examples</u> retrieves several training triples that share the same relation (*playfor*) as the current query to serve as demonstration prompts, aiding the LLM in learning relational patterns. For instance: (*Jason Tatum*, *playfor*, *Celtics*), (*LeBron James*, *playfor*, *Lakers*), and (*Wembanyama*, *playfor*, *Spurs*).

<u>Context Distillation</u> provides semantic contexts by extracting textual descriptions of the head entity (*Stephen Curry*) and the relation (*playfor*) from sources such as Wikidata. If such descriptions are unavailable, they are generated via an LLM.

<u>Auto Path Selection</u> dynamically retrieves one-hop and multi-hop neighbors of the head entity (e.g., Stephen Curry) based on its connectivity, using fewer hops for dense entities and more for sparse ones. An attention mechanism ranks neighbor relevance, and only the most informative paths are retained to provide structural context, such as (*Curry, team_location, San Francisco, venue_of, Chase Center*) and (*Draymond Green, teammates, Curry*).

<u>Candidates Generation</u> refers to the top-M candidate entities (e.g., [*Lakers*, *Celtics*, *Warriors*, ...]) produced by the FKGC model. This list reflects the initial judgment of the meta-learning and is retained in the prompt to guide the LLM toward more refined predictions.

3.5.2 Knowledge Selector. To improve the efficiency and effectiveness of instruction fine-tuning, we propose a Knowledge Selector for constructing high-quality instruction data. Specifically, we introduce two complementary sampling strategies to evaluate the importance of relations in the training set: Diversity-aware Sampling and Centrality-aware Sampling.

<u>Diversity-aware Sampling</u> To ensure that the instruction data covers diverse relational patterns in the knowledge graph, we first apply the K-Means algorithm to cluster all relation embeddings. Specifically, given a set of relation embeddings $\{r_1, r_2, \ldots, r_n\}$ from the training set, we partition them into $\mathcal{K}$ clusters, each represented by a centroid μ_k. Within each cluster C_k, the relation closest to the centroid(r_k^*) is selected as the representative.

$$r_k^* = \arg\min_{r \in C_k} \|r - \mu_k\|_2 . \tag{21}$$

This method automatically selects the most representative relations, ensuring that the instruction data covers diverse semantic patterns while avoiding redundancy.

<u>Centrality-aware Sampling</u> To enhance the model's ability to learn core knowledge, we propose a relation-level PageRank-based centrality metric. Unlike traditional entity-centric centrality measures, we construct a graph where nodes represent relations and edges are formed based on shared entities. The knowledge centrality score is then computed for each relation as follows:

$$KC(r) = \frac{1-d}{d} + d \cdot \sum_{r' \in N(r)} \frac{KC(r')}{|N(r')|}, \tag{22}$$

where $KC(r)$ denotes the knowledge centrality score of relation r, d is the damping factor (typically set to 0.85), $N(r)$ represents the set of neighboring relations of r, and $|N(r')|$ is the number of neighbors of relation r'. We select the top $T\%$ of relations with the highest centrality scores to ensure that the training data covers the most critical and influential knowledge in the knowledge graph.

Finally, we take the union of the results from diversity-aware and centrality-aware sampling, ensuring that the selected relations both cover diverse semantic patterns and reflect core knowledge in the graph. This strategy not only reduces the size of the instruction data, but also guides the model to focus on essential knowledge.

3.5.3 Re-ranker via Instruction Fine-tuning. After retrieving the relevant instruction contexts and distilling high-quality knowledge, we fine-tune a large language model (LLM) specifically for few-shot knowledge graph completion (FKGC). Given the carefully constructed prompts $P(q)$, which incorporate both structural and semantic context, along with the top-M candidates generated by the meta-learning model, LLM is fine-tuned to refine these candidates and ultimately rank the top-10 most plausible tail entities in order.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets. We evaluate our LGC-CR on three benchmark datasets: NELL-One, Wiki-One [39], and FB15K-One [40]. Following established protocols [20, 42], we select relations with 50–500 triples for few-shot tasks, while treating the remaining relations and their triples as the background knowledge graph. Based on this setup, we employ Deepseek-R1[8] to generate textual descriptions for each entity and relation by querying it for relevant information, and evaluate their quality by comparing them with corresponding Wikipedia entries for semantic consistency. Detailed dataset statistics are provided in Table 2.

Table 2: Statistics of datasets.

Dataset	Ent	Triple	Rel	Task Train/Valid/Test	Avg. Desc. Length (E/R)
NELL	68,545	181,109	358	67(51/5/11)	30.5/10.2
Wiki	4,838,244	5,859,240	822	183(133/16/34)	24.7/8.4
FB15K	14,478	309,621	237	45(32/8/5)	19.6/7.1

4.1.2 Baselines. We compare our model with two types of FKGC baselines. **(1) Metric-based:** GMatching [39], FAAN [30], CIAN [19], TransAM [23], APINet [20], SuperRL [11], MVSE [26], HNII [21], and NFAA [7]. **(2) Optimization-based:** MetaR [3], FSRL [42], and GANA [27]. These methods focus on local structures and relational semantics for better embeddings, without incorporating knowledge from LLM. Traditional KGE models are omitted due to significantly worse performance than the few-shot baselines [27].

4.1.3 Implementation Details. For fair comparison, we use TransE-pretrained entity and relation embeddings. For each dataset, we set the number of neighbors to 100, and fix the filtering parameter β at 1.0. We set the neighbor hop at 2, the compare network weight ω at 100, the global score weight θ at 0.6, the number of negative samples at 8, the temperature τ at 0.1, and the margin γ at 5.0. The Adam optimizer [15] is used with learning rate of $1e^{-4}$ for NELL-One, $2e^{-4}$ for Wiki-One, and $8e^{-5}$ for FB15K-One. The hyperparameter λ is set to 0.3 for 3-shot and 0.06 for 5-shot. During refinement, we consider $M = 20$ candidate answers. Diversity sampling employs 3/10/5 clusters for NELL/Wiki/FB15K, respectively, while centrality sampling retains the top 30% of relations. For instruction tuning, we adopt Deepseek-R1-Distill-Qwen-14B as the LLM and apply LoRA [12] for finetuning ($r = 8$, $alpha = 16$). All experiments are conducted using PyTorch on an NVIDIA RTX A800 GPU.

4.2 Main Results

4.2.1 Performance Comparison. Table 3 compares LGC-CR and its two variants, LGC-FKGC (using only the enhanced meta-learning model) and LGC+Deepseek-R1 (employing the LLM for refinement without fine-tuning), against the baselines on three benchmark datasets under 3-shot and 5-shot settings. From the experimental results, we have the following observations:

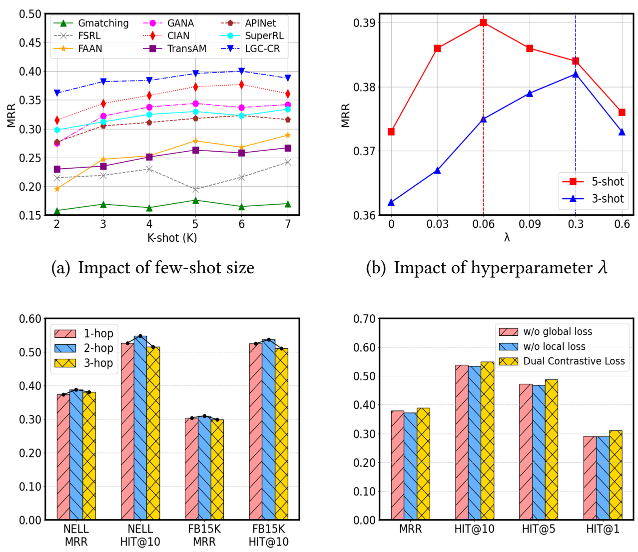
- LGC-CR outperforms the state-of-the-art methods on three benchmark datasets. Specifically, compared with the best-performing baseline, LGC-CR achieves remarkable improvements of 8.1%, 21.7 %, and 20.6% in Hit@1 on Nell-One, Wiki-One, and FB15K-One datasets, respectively, indicating that LGC-CR is more likely to rank correct triplets at the top position.
- LGC-FKGC is also competitive, particularly under 3-shot setting. This is because when few triplets are available, our method can obtain stable and denoised global neighborhood information to make up for the performance degradation caused by the lack of support set. Compared to NFAA, LGC-FKGC performs better, which benefits from our dual contrastive learning, enabling better integration of neighborhood information from different views. Moreover, our method outperforms metric-based baseline methods (e.g., CIAN), thanks to the fusion similarity network, which is able to construct a more discriminative nonlinear metric space. Additionally, the performance improvement on NELL is greater than on Wiki, likely due to differences in graph density. In dense graphs like NELL, FilGAT can filter out many dissimilar neighbors and focus on a few relevant ones for effective feature aggregation. In contrast, Wiki has sparser neighborhoods, where fewer irrelevant neighbors need to be identified.
- With the addition of LLM refinement, our method significantly outperforms embedding-based meta-learning methods. This improvement is attributed to the strong semantic understanding of the LLM, which mitigates the impact of spurious relation patterns. However, without fine-tuning, this enhancement is less evident, especially on NELL dataset, which contains many specialized entities (e.g., apple001). The LLM struggles to comprehend such domain-specific knowledge in the KG, underscoring the importance of the knowledge selector and fine-tuning, which explicitly injects filtered knowledge to further boost model performance.

4.2.2 Ablation Study. We perform ablation studies on the NELL-One and Wiki-One datasets to assess the contribution of each component in LGC-CR. Results presented in Table 4 demonstrate that each module plays a vital role in the overall performance. Specifically, (1)When removing the knowledge selector (*w/o.KS*), we observe noticeable performance degradation (NELL-One: Hit@1 ↓1.9%; Wiki-One: Hit@1 ↓2.9%). This confirms that selecting high-diversity, high-centrality facts helps reduce noise from marginal relations. Further removing the fine-tuning module (*w/o. KS&sft*) results in more significant performance drops (NELL-One: Hit@1 ↓5.3%; Wiki-One: Hit@1 ↓7.8%), indicating that knowledge injection through fine-tuning enhances LLM performance and contributes to better refinement results. Additionally, eliminating the refinement module (*w/o. CR*) leads to substantial performance declines (Wiki-One: Hit@1 ↓23.3%), demonstrating its ability to improve meta-learned outputs via strong semantic understanding. (2) Removing

Table 3: Comparison of state-of-the-art methods on Nell-One, Wiki-One, and FB15K-One.

Model	Nell-One						Wiki-One						FB15K-One		
	MRR		Hit@5		Hit@1		MRR		Hit@5		Hit@1		MRR	Hit@5	Hit@1
	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	5-shot	5-shot	5-shot
GMatching(2018)[39]	-	0.176	-	0.233	-	0.113	-	0.263	-	0.337	-	0.197	0.189	0.274	0.101
MetaR (2019)[3]	0.245	0.261	0.360	0.350	0.144	0.168	0.317	0.323	0.379	0.385	0.261	0.270	0.203	0.291	0.107
FSRL(2020)[42]	0.219	0.195	0.296	0.279	0.139	0.108	0.102	0.113	0.131	0.135	0.050	0.056	0.223	0.364	0.102
FAAN(2020)[30]	0.247	0.279	0.309	0.364	0.183	0.200	0.298	0.341	0.368	0.395	0.228	0.281	0.259	0.424	0.178
GANA(2021)[27]	0.322	0.344	0.432*	0.437	0.225	0.246	0.331	0.351	0.389	0.407	0.283	0.299	0.209	0.334	0.107
CIAN(2022)[19]	0.344*	0.373*	0.417	0.453	0.266*	0.294*	0.358	0.383	0.438	0.453	0.284	0.318	0.248	0.426	0.144
TransAM(2023)[23]	0.235	0.263	0.361	0.311	0.175	0.205	0.315	0.330	0.345	0.405	0.273	0.258	-	-	-
APINet(2023)[20]	0.305	0.318	0.405	0.412	0.208	0.225	0.342	0.347	0.419	0.428	0.283	0.297	-	-	-
SuperRL(2024)[11]	0.312	0.330	0.404	0.441	0.223	0.234	0.359*	0.388*	0.446*	0.458*	0.297*	0.320*	-	-	-
MVSE(2024)[26]	0.302	0.332	0.405	0.465*	0.178	0.211	0.349	0.351	0.382	0.422	0.295	0.296	0.285*	0.458*	0.195*
HNII(2024)[21]	-	0.365	-	0.453	-	0.283	-	0.352	-	0.457	-	0.315	-	-	-
NFAA(2025)[7]	0.303	0.332	0.368	0.409	0.233	0.263	0.332	0.341	0.384	0.416	0.285	0.282	-	-	-
LGC-FKGC	0.382	0.390	0.467	0.488	0.304	0.311	0.362	0.377	0.458	0.473	0.301	0.304	0.310	0.469	0.208
LGC+DeepSeek-R1	-	0.392	-	0.473	-	0.322	-	0.486	-	0.525	-	0.459	0.396	0.518	0.300
LGC-CR	**0.428**	**0.440**	**0.515**	**0.523**	**0.342**	**0.375**	**0.544**	**0.567**	**0.573**	**0.606**	**0.511**	**0.537**	**0.474**	**0.566**	**0.401**

Best results are **bolded**, runner-up results are underlined, and * indicates the SOTA baseline metrics.
Missing metrics are due to unavailability in the original papers.



(a) Impact of few-shot size

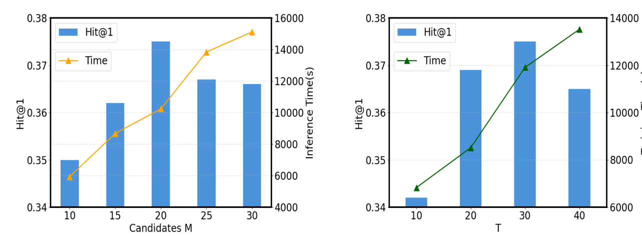
(b) Impact of hyperparameter λ

(c) Impact of neighbor hop

(d) Impact of Contrastive Strategies

Figure 4: Parameter Sensitivity Analysis of LGC-FKGC.

the dual contrastive learning (*w/o. CR&DCL*) causes notable performance declines, verifying its effectiveness in aligning local-global views and enhancing model robustness. Replacing our fusion similarity network with a single dot-product metric (*w/o. CR&FS*) yields inferior results, suggesting that relying solely on a single metric can introduce similarity bias, whereas integrating Mahalanobis distance and compare network helps construct a more discriminative metric space. Furthermore, removing the local-global contrast module (*w/o. CR&LGC*) leads to additional performance degradation, indicating



(a) Impact of number of candidates

(b) Impact of Knowledge Selector ratio

Figure 5: Impact of Candidate Entities and Knowledge Selector Ratio on LGC-CR Performance and Time Cost.

that rich high-order neighbor information and noise filtering are crucial to model effectiveness. (3) *LLM-only (shuffle)* refers to using the LLM to process randomly ordered candidate entities, which causes a notable drop in performance compared to the full model. This indicates that the initial meta-learning stage reduces the ranking difficulty. *LLM-only (generate)* denotes prompting the LLM to directly generate an answer and ground it to candidate entities via similarity, but yields the worst performance, suggesting that the alignment process introduces inevitable errors. The poor performance of LLM-only variants further validates the effectiveness of our integrated approach.

4.3 Further Analysis

4.3.1 **Impact of few-shot size.** We conducted experiments on LGC-CR while varying the few-shot size K on Nell-One. As depicted in Figure 4(a), (1) When K is small, the performance of LGC-CR is significantly better than other methods. This is because, with a limited number of triplets available, our method can effectively leverage stable and denoised global neighborhood information to

Table 4: Ablation study results on Nell-One and Wiki-One.

Models	Nell-One		Wiki-One	
	MRR	Hit@1	MRR	Hit@1
Full Model	**0.440**	**0.375**	**0.567**	**0.537**
w/o. KS	0.420	0.356	0.538	0.508
w/o. KS&sft	0.392	0.322	0.486	0.459
w/o. CR	0.390	0.311	0.377	0.304
w/o. CR&FS	0.384	0.302	0.364	0.295
w/o. CR&DCL	0.376	0.286	0.359	0.288
w/o. CR&LGC	0.353	0.269	0.342	0.280
LLM-only(shuffle)	0.267	0.282	0.336	0.269
LLM-only(generate)	0.110	0.135	0.110	0.226

compensate for the performance drop caused by the lack of support set. (2) The performance of LGC-CR fluctuates slightly across different values of K, maintaining relatively stable results. This stability is attributed to the use of residual connections and dual contrastive learning strategies, which help stabilize the training process and enhance the model's generalization ability, enabling it to handle various few-shot scenarios.

4.3.2 **Impact of hyperparameter λ.** We explored the impact of the dual contrastive learning loss weight λ on performance under different settings (3-shot and 5-shot). The values of λ were selected from the set $\{0, 0.03, 0.06, 0.09, 0.3, 0.6\}$. As shown in Figure 4(b), in the 3-shot setting, the model's performance (MRR) reached its peak when $\lambda = 0.3$. In the 5-shot setting, the performance was maximized when $\lambda = 0.06$. This indicates that the impact of the contrastive learning loss on model performance is more significant when the support size is smaller. With fewer support set triplets, the model is more prone to overfitting, and aligning multi-view neighborhood features can enhance the model's generalization ability.

4.3.3 **Impact of neighbor hop**. We analyzed the model's performance on NELL-One and FB15K-One datasets across different neighbor hop counts (1-hop, 2-hop, 3-hop). As shown in Figure 4(c), performance improves on both datasets when increasing from 1-hop to 2-hop, indicating that appropriately increasing neighbor hop count helps capture richer neighborhood information. However, excessive hop counts (e.g., 3-hop) lead to performance decline: MRR dropped from 0.390 to 0.381 on NELL-One and from 0.310 to 0.299 on FB15K-One, suggesting that broader neighbor ranges may introduce additional noise. Notably, 3-hop performance on NELL-One remains relatively high, while it's significantly lower on FB15K-One, possibly due to FB15K-One's smaller size making it more susceptible to training noise.

4.3.4 **Impact of Contrastive Learning Strategies**. We compare three contrastive learning strategies on the NELL-one dataset: Dual Contrastive Learning, using only the local view as the anchor (w/o. global loss), and using only the global view as the anchor (w/o. local loss). The results shown in Figure 4(d) denote that Dual Contrastive Learning outperforms the single contrastive learning strategies across all metrics, especially in Hit@1. This is because single contrastive learning is prone to overfitting the noise in the current view. In contrast, Dual Contrastive Learning simultaneously optimizes both views, meaning that even if one view is incomplete or noisy, the other can still provide valuable complementary information, enhancing the model's robustness.

4.3.5 **Impact of number of candidates**. We investigated the impact of the number of candidate entities M generated by meta-learning on model performance and inference time. The experimental results are shown in Figure 5(a). First, as M increases, the training time grows linearly, which is intuitive because an increase in M directly leads to longer prompt content. Next, focusing on the performance of LGC-CR, it was found that setting M to 20 yields the best performance on the NELL-One dataset. However, further increasing M leads to a decrease in performance. This result indicates that blindly increasing the number of candidate entities does not necessarily improve performance.

4.3.6 **Impact of the Knowledge Selector ratio**. We investigated the impact of the knowledge selector ratio T on the performance and training time of LGC-CR. The experimental results, shown in Figure 5(b), lead to the following observations: (1) As T increases, the training time grows linearly, since a larger T means more training data, resulting in higher computational cost. (2) When $T = 30$, the model performs best, indicating that the knowledge selector effectively identifies the most central training data, enhancing the model's generalization. (3) When $T = 40$, performance drops. Although the data volume increases, more noisy or irrelevant relations are introduced, weakening the fine-tuning effect. (4) When $T = 10$, Hit@1 is lower, as insufficient training data prevents the model from learning essential knowledge. Thus, too little data hinders model adaptation, while too much data may introduce noise.

4.4 Case Study

4.4.1 Visualization. We conducted a case study by selecting two specific relations for visualization, as shown in Figure 6. The figure demonstrates that our model effectively distinguishes between the representations of positive and negative entity pairs. Compared to baseline methods such as CIAN [19] and MVSE [26], the separation is noticeably clearer. For instance, under the relation *'animalssuchasinvertebrate'*, the positive and negative pairs are clearly separable. This observation indicates that our model excels at producing highly distinctive and well-separated representations for different types of entity pairs.

4.4.2 Relation Pattern Bias Analysis. To analyze the improvement of our method on spurious pattern bias across different relations, we conduct a case study comparing the top-5 candidate entities produced by the meta-learning model and after LLM-guided refinement, as shown in Table 5. For the *teamcoach* relation, the LGC-FKGC model suffers from high-frequency patterns (e.g., 'NBA team → famous coach', 'same-conference teams → common star coaches'), ranking hub entities such as Phil Jackson and Gregg Popovich ahead of the true but long-tail head coach (Jason Kidd). Notably, LLM refinement also considers temporal relevance, placing Jason Kidd before Rick Carlisle. For the *location of discovery* relation, the model exhibits a 'discoverer–site' bias: if an artifact's discoverer has found other items in a location (e.g., Siselen), the model tends to assume the artifact was also discovered there, thus over-ranking that location.

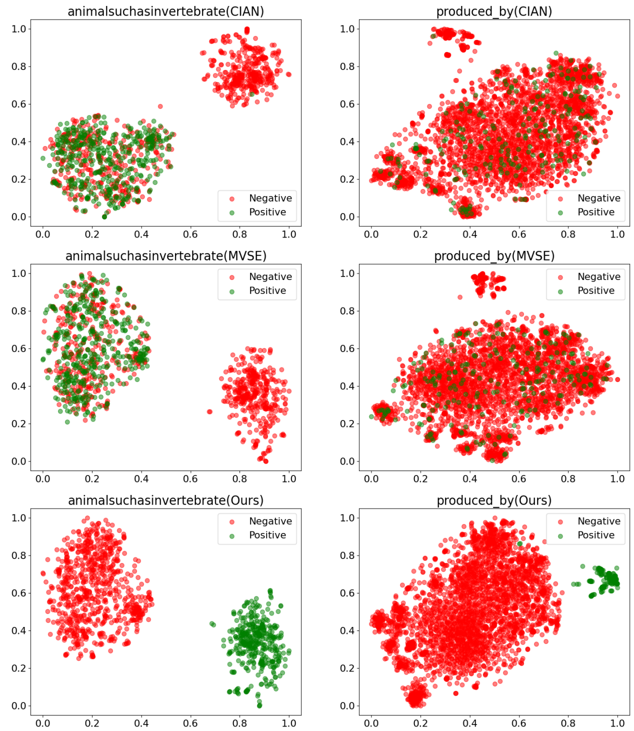


Figure 6: Embedding visualization of entity pairs.

Table 5: Case study of Top-5 candidate rankings for two relation types. Correct entities are in bold.

Query Task	LGC-FKGC	LGC-CR
(dallas_mavericks, teamcoach, ?) *True:* coach:jason_kidd	phil_jackson gregg_popovich scott_skiles rick_carlisle **jason_kidd**	**jason_kidd** rick_carlisle gregg_popovich scott_skiles marc_iavaroni
(possible_agrippina_major, location_of_discovery, ?) *True:* béziers	siselen müllheim **béziers** labastide_marnhac isarrnig	**béziers** siselen müllheim labastide_marnhac river_ness

Our LLM-Guided Refinement integrates textual evidence and factual knowledge to promote the correct answer to the top, effectively mitigating the pattern bias of meta-learning methods.

5 Conclusion

In this paper, we introduced LGC-CR, a novel few-shot knowledge graph completion framework that integrates an enhanced meta-learning model with LLM knowledge. LGC-CR includes a local-global contrastive network that aggregates local information through cross-attention and captures global neighborhoods via a hierarchical module, effectively enriching high-order neighborhood information while flexibly filtering noise. To effectively fuse these complementary multi-view neighborhoods, we introduced a dual contrastive learning module to ensure consistency. Additionally, we designed a fusion similarity network to construct a more discriminative non-linear metric space. Finally, we incorporated LLM refinement, which retrieves relevant contexts and selects diverse, knowledge-centric facts to fine-tune LLMs for optimizing meta-learning results. Extensive experiments demonstrate that LGC-CR achieves significant improvements and exhibits more robust performance than state-of-the-art baselines, with Hit@1 improvements of 8.1%, 21.7%, and 20.6% on NELL, Wiki, and FB15K, respectively.

Acknowledgments

The research was partially supported by “Pioneer” and “Leading Goose” R&D Program of Zhejiang 2023C01029, the Plans for Major Provincial Science&Technology Projects No.202303a07020006, and the China National Natural Science Foundation with No.62132018.

GenAI Usage Disclosure

During the preparation of this work, the authors used ChatGPT [1] to polish the paper and subsequently reviewed and edited the content. They take full responsibility for the final publication.

References

[1] Sakher Khalil Alqaaidi and Krzysztof J Kochut. 2024. Relations Prediction in Knowledge Graph Completion using Large Language Models. In *Proceedings of the 2024 8th International Conference on Information System and Data Mining*. 122–127.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[3] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. *arXiv preprint arXiv:1909.01515* (2019).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[5] Huifang Du, Zhongwen Le, Haofen Wang, Yunwen Chen, and Jing Yu. 2022. COKG-QA: Multi-hop question answering over COVID-19 knowledge graphs. *Data Intelligence* 4, 3 (2022), 471–492.

[6] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. 2017. Predicting completeness in knowledge bases. In *Proceedings of the tenth acm international conference on web search and data mining*. 375–383.

[7] Hongfang Gong, Yingjing Ding, and Minyi Ma. 2025. A few-shot knowledge graph completion model with neighbor filter and affine attention. *IEEE Access* (2025).

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[9] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3549–3568.

[10] Yuxiang Guo, Shuanghong Shen, Qi Liu, Zhenya Huang, Linbo Zhu, Yu Su, and Enhong Chen. 2024. Mitigating Cold-Start Problems in Knowledge Tracing with Large Language Models: An Attribute-aware Approach. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 727–736.

[11] Jiewen Hou, Tianxing Wu, Jingting Wang, Shuang Wang, and Guilin Qi. 2024. Supervised Relational Learning with Selective Neighbor Entities for Few-Shot Knowledge Graph Completion. In *International Semantic Web Conference*. Springer, 144–161.

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[1] chatgpt.com

[13] Yubo Huang and Guosun Zeng. 2024. RD-P: A Trustworthy Retrieval-Augmented Prompter with Knowledge Graphs for LLMs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 942–952.

[14] Sakher Khalil Alqaaidi and Krzysztof Kochut. 2024. Relations Prediction for Knowledge Graph Completion using Large Language Models. *arXiv e-prints* (2024), arXiv–2405.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Dawei Li, Zhen Tan, Tianlong Chen, and Huan Liu. 2024. Contextualization distillation from large language model for knowledge graph completion. *arXiv preprint arXiv:2402.01729* (2024).

[17] Jinlin Li, Zikang Wang, Linjing Li, and Daniel Dajun Zeng. 2024. Relation Adaptive Representation Learning Based on Factual Information Interaction for One-Shot Knowledge Graph Completion. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[18] 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 *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Vol. 271494703.

[19] Yuling Li, Kui Yu, Xiaoling Huang, and Yuhong Zhang. 2022. Learning inter-entity-interaction for few-shot knowledge graph completion. In *Proceedings of the 2022 conference on empirical methods in natural language processing*. 7691–7700.

[20] Yuling Li, Kui Yu, Yuhong Zhang, Jiye Liang, and Xindong Wu. 2023. Adaptive prototype interaction network for few-shot knowledge graph completion. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[21] Zihao Li, Lin Feng, Lingxiao Xu, Qiuping Shuai, and Ling Yue. 2024. Exploring Hierachical Neighbor Information Interaction for Few-Shot Knowledge Graph Completion. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[22] Zhuofeng Li, Haoxiang Zhang, Qiannan Zhang, Ziyi Kou, and Shichao Pei. 2024. Learning from novel knowledge: Continual few-shot knowledge graph completion. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1326–1335.

[23] Yi Liang, Shuai Zhao, Bo Cheng, and Hao Yang. 2023. TransAM: Transformer appending matcher for few-shot knowledge graph completion. *Neurocomputing* 537 (2023), 61–72.

[24] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering* 35, 1 (2021), 857–876.

[25] Yang Liu, Zequn Sun, Guangyao Li, and Wei Hu. 2022. I know what you do not know: Knowledge graph embedding via co-distillation learning. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 1329–1338.

[26] Ruixin Ma, Hao Wu, Xiaoru Wang, Weihe Wang, Yunlong Ma, and Liang Zhao. 2024. Multi-view semantic enhancement model for few-shot knowledge graph completion. *Expert Systems with Applications* 238 (2024), 122086.

[27] Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 213–222.

[28] Janna Omeliyanenko, Albin Zehe, Andreas Hotho, and Daniel Schlör. 2023. CapsKG: enabling continual knowledge integration in language models for automatic knowledge graph completion. In *International Semantic Web Conference*. Springer, 618–636.

[29] Diego Sanmartin. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035* (2024).

[30] Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive attentional network for few-shot knowledge graph completion. *arXiv preprint arXiv:2010.09638* (2020).

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

[32] Yu Song, Mingyu Gui, Kunli Zhang, Zexi Xu, Dongming Dai, and Dezhi Kong. 2024. Relational multi-scale metric learning for few-shot knowledge graph completion. *Knowledge and Information Systems* 66, 7 (2024), 4125–4150.

[33] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).

[34] Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. 2021. MoCL: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 3585–3594.

[35] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[36] Yanbin Wei, Qiushi Huang, James T Kwok, and Yu Zhang. 2024. Kicgpt: Large language model with knowledge in context for knowledge graph completion. *arXiv preprint arXiv:2402.02389* (2024).

[37] Tianxing Wu, Haofen Wang, Cheng Li, Guilin Qi, Xing Niu, Meng Wang, Lin Li, and Chaomin Shi. 2020. Knowledge graph construction from multiple online encyclopedias. *World Wide Web* 23 (2020), 2671–2698.

[38] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. 2022. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022*. 1070–1079.

[39] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. *arXiv preprint arXiv:1808.09040* (2018).

[40] Jingwen Xu, Jing Zhang, Xirui Ke, Yuxiao Dong, Hong Chen, Cuiping Li, and Yongbin Liu. 2021. P-INT: A path-based interaction model for few-shot knowledge graph completion. In *Findings of the association for computational linguistics: EMNLP 2021*. 385–394.

[41] Yanlin Yang, Zhonglin Ye, Haixing Zhao, and Lei Meng. 2023. A novel link prediction framework based on gravitational field. *Data Science and Engineering* 8, 1 (2023), 47–60.

[42] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3041–3048.

[43] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web* 27, 5 (2024), 58.