

Bridging Text and Structure via Latent Query Refinement for Knowledge Graph Completion

Chunmiao Yu^{1,2}, Zikang Wang¹, Tianyi Luo¹, Jiaojiao Wang¹, Zhidong Cao^{1*}

¹State Key Laboratory of Multimodal Artificial Intelligence Systems

Institute of Automation, Chinese Academy of Sciences, Beijing, China

²School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

{yuchunmiao2022, wangzikang2016, luotianyi2017}@ia.ac.cn

{jjiaojiao.wang, zhidong.cao}@ia.ac.cn

Abstract

Knowledge graph completion(KGC) aims to infer missing facts from observed relational triples, which demands jointly modeling textual semantics and relational structure. Existing structure-based methods capture graph topology but often struggle with sparse entities and long-tail relations, while text-based methods exploit entity and relation descriptions but weakly model relational neighborhoods. In this paper, we propose LQ-KGC, a latent-query refinement framework that enables iterative interaction between semantic inference and structural reasoning for KGC. Instead of treating all missing triples as latent variables, LQ-KGC constructs a restricted set of semantically plausible latent queries and alternates between two refinement steps: the language model predicts latent tail distributions from textual descriptions, while the graph model incorporates high-confidence pseudo tails as pseudo edges for relation-aware message passing. Meanwhile, structure-aware graph predictions are distilled back into the language model to refine semantic representations. Experiments on WN18RR and FB15k-237 demonstrate that LQ-KGC consistently improves over text-based and graph-based baselines, with additional analyses highlighting the effectiveness of pseudo-edge augmentation and latent-query construction.

1 Introduction

Knowledge graphs (KGs) store factual knowledge as relational triples (h, r, t) and support many knowledge-intensive applications, including question answering (Yasunaga et al., 2021), information retrieval (Liu et al., 2018), recommendation (Wang et al., 2021b), and domain-specific reasoning in biomedical (Zhang et al., 2025) and e-commerce (Li et al., 2020) scenarios. However, real-world KGs are inherently incomplete, since

knowledge acquisition is costly and factual information continuously evolves. This incompleteness limits downstream applications and motivates the task of knowledge graph completion (KGC), which aims to infer missing facts from observed triples.

Existing KGC methods mainly fall into structure-based and text-based approaches. Structure-based methods, including translational models, semantic matching models, and graph neural networks (GNNs), learn entity and relation representations from observed triples and relational neighborhoods. These approaches effectively capture graph topology and relation patterns, but their performance heavily depends on observed edges and often degrades for sparse entities and long-tail relations. In contrast, text-based methods leverage entity descriptions and pretrained language models (PLMs) to provide semantic evidence beyond graph topology. Recent PLM-based methods such as SimKGC (Wang et al., 2022a) achieve strong performance by encoding queries and candidate entities into a shared semantic space. Nevertheless, these methods primarily rely on local textual matching and weakly exploit global relational structure.

Textual semantics and graph structure provide complementary signals for KGC, yet effectively coupling the two remains challenging. Existing hybrid approaches typically rely on static alignment or one-way structural injection, resulting in limited bidirectional interaction between semantic inference and structural reasoning. In particular, semantic evidence inferred by language models is rarely incorporated into graph reasoning as structural context, while graph structure is often underutilized for refining semantic predictions.

In this paper, we propose LQ-KGC, a latent-query refinement framework for text-enhanced KGC. Rather than modeling all missing triples as latent variables, LQ-KGC constructs a restricted set of semantically plausible latent queries and alternates between semantic and structural refine-

*Corresponding author.

ment. In semantic refinement, the language model predicts latent tail distributions from textual descriptions and receives structure-aware supervision from the graph model. In structural refinement, high-confidence semantic predictions are injected as pseudo edges, enabling relation-aware message passing over both observed and inferred connections. Through iterative refinement, semantic inference and structural reasoning mutually enhance each other. To mitigate noisy pseudo-label propagation, we further introduce a query construction strategy that selects semantically compatible and distributionally balanced latent queries, improving the quality and diversity of pseudo completions, especially for sparse entities and long-tail relations.

We evaluate LQ-KGC on WN18RR and FB15k-237 under the standard filtered ranking protocol. Experimental results show that LQ-KGC consistently outperforms strong text-based and graph-based baselines, achieving relative MRR improvements of 2.2% over SimKGC on WN18RR and 9.1% over CompGCN on FB15k-237. Additional analyses further demonstrate the effectiveness of pseudo-edge augmentation and latent-query construction for long-tail completion.

Our contributions are summarized as follows:

- We propose LQ-KGC, a latent-query refinement framework for text-enhanced KGC, which enables iterative interaction between semantic inference and structural reasoning.
- We design a bidirectional refinement mechanism and a latent query construction strategy, where graph predictions supervise semantic refinement, LM-derived pseudo tails augment graph message passing, and semantically plausible balanced queries reduce noisy pseudo-label exchange.
- We conduct experiments and ablation studies on WN18RR and FB15k-237 to evaluate the effectiveness of LQ-KGC, and analyze the role of latent query construction, pseudo-edge augmentation, and text-graph refinement.

2 Related Work

2.1 Structure-based KGC Methods

Structure-based KGC methods infer missing facts from observed triples by learning entity and relation representations from graph structure. Early embedding models such as TransE (Bordes et al.,

2013), TransH (Wang et al., 2014), TransD (Ji et al., 2015) and RotatE (Sun et al.) model relations as geometric transformations over embeddings. Semantic matching and neural decoders, including DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2017), and ConvE (Dettmers et al., 2018) further improve expressiveness by capturing more relation patterns. More recently, GNN-based methods such as R-GCN (Schlichtkrull et al., 2018), CompGCN (Vashishth et al., 2019), and HittER (Chen et al., 2021) exploit relational neighborhoods through message passing or hierarchical contextualization. These methods are effective at capturing graph topology, but their performance depends heavily on observed connectivity and therefore remains sensitive to sparse entities and long-tail relations.

2.2 Text-based KGC Methods

Text-based KGC methods use entity names, relation labels, and descriptions as semantic evidence. Sequence-level methods, such as KG-BERT (Yao et al., 2019) and multi-task PLM models (Kim et al., 2020), encode textualized triples and predict plausibility, but are costly for all-entity ranking. Generative methods such as KGT5 (Saxena et al., 2022) and KG-S2S (Chen et al., 2022) reformulate KGC as text-to-text prediction, but require constrained decoding and entity canonicalization. More scalable retrieval-style methods, including SimKGC (Wang et al., 2022a) and LMKE (Wang et al., 2022b), encode queries and candidate entities separately and optimize contrastive objectives over positive and negative entities. GHN (Qiao et al., 2023) further improves contrastive text-based KGC by using a sequence-to-sequence generator to produce semantically close hard negatives. These methods provide useful semantic generalization, especially for sparse or unseen entities, yet they primarily rely on local textual compatibility.

2.3 Text-structure Fusion Methods

Text-structure fusion methods seek to combine semantic evidence from language models with relational evidence from KG topology. Joint objective methods, such as KEPLER (Wang et al., 2021c) and StAR (Wang et al., 2021a), align textual representations with KG embedding or structure-aware scoring objectives. They improve over purely textual modeling, but the interaction between text and structure is usually fixed after training or handled by score-level fusion. Prompt- and adapter-based

methods, including CSProm-KG (Chen et al., 2023) and KoPA (Zhang et al., 2024), inject structural signals into PLMs or LLMs through lightweight interfaces. LLM-based pipelines (Chen et al., 2024) further use language models for alignment between text and graph representations. SAT (Liu et al., 2025) aligns graph embeddings with natural language space and performs structure-aware instruction tuning for LLM-based KGC. Memory-KGC (Li et al., 2026) also follows this fusion paradigm by combining PLM-based entity encodings with subgraph-driven structural memory.

EM-style and alternating methods provide closer points of comparison. ExpressGNN (Zhang et al., 2020) combines logical rules with GNN-based reasoning, and RNNLogic (Qu et al., 2021) alternates between rule generation and KG reasoning; both focus on rule-based structural inference. DynaSemble (Nandi et al., 2024) dynamically ensembles textual and structure-based KGC models at inference time. GLEM (Zhao et al.) exchanges pseudo labels between language and graph models, but targets node-level prediction on text-attributed graphs rather than relation-aware entity ranking in KGC. LQ-KGC differs by performing training-time pseudo-label exchange between a text retriever and a relation-aware GNN: semantic predictions are converted into typed pseudo edges, while graph predictions provide structural supervision for semantic refinement.

3 Method

Figure 1 illustrates the overall framework of LQ-KGC. Inspired by EM-style iterative optimization, LQ-KGC alternates between semantic refinement and structural refinement over a restricted set of latent queries. We use the terms E-step and M-step as an optimization analogy over sampled latent queries, rather than as exact EM over the full set of unobserved triples. The language model predicts semantic distributions over candidate tails from textual descriptions, while the graph model performs structure-aware reasoning over observed and pseudo-augmented graphs. The two branches exchange supervision asymmetrically: pseudo edges from the LM primarily strengthen graph learning, whereas graph-to-text distillation acts as a structure-aware regularizer. In the following, we describe the problem formulation, the approximate refinement objective, and the two refinement steps in detail.

3.1 Problem Formulation

A knowledge graph is denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ is the set of observed factual triples. Each triple (h, r, t) indicates that relation r holds between head entity h and tail entity t . In the text-enhanced setting, each entity $e \in \mathcal{E}$ is associated with textual information s_e , such as its name and description, and each relation $r \in \mathcal{R}$ may also have a textual description s_r .

The goal of knowledge graph completion (KGC) is to infer missing facts from the partially observed graph. In this work, we focus on tail prediction. Given an incomplete query $(h, r, ?)$, the model estimates the plausibility of every candidate entity $t \in \mathcal{E}$ and ranks the correct tail entity highly. Equivalently, KGC can be formulated as learning a conditional distribution $p(t \mid h, r, \mathcal{G})$ or a scoring function $f(h, r, t)$ such that true triples receive higher scores than implausible triples.

3.2 EM-inspired Latent Query Refinement

LQ-KGC alternates between two tractable surrogate updates over sampled latent queries. For each query q , the text model defines a semantic distribution $q_\phi(z_q \mid q, \mathcal{D})$, while the graph model provides a structure-induced distribution $\tilde{p}_\theta(z_q \mid q, \mathcal{G}_{\text{obs}})$. These two models exchange supervision in both directions: the graph model provides soft structural guidance for semantic refinement, and the text model contributes high-confidence pseudo tails for structural refinement. This alternating procedure allows semantic evidence and graph structure to improve each other gradually. In the following, we describe the approximate E-step and approximate M-step in detail.

3.3 E-step: Semantic Refinement

In the E-step, the structure-based model is fixed, and the text-based variational distribution q_ϕ is updated to infer the latent tail variable z_q for each sampled query $q = (h, r) \in \mathcal{Q}$. We instantiate q_ϕ with a bi-encoder like (Wang et al., 2022a), where the query (h, r) and candidate entity t are encoded by two language encoders with the same architecture. The model computes a semantic matching score

$$s_\phi(h, r, t) = \text{sim}_\phi(h, r, t; \mathcal{D}),$$

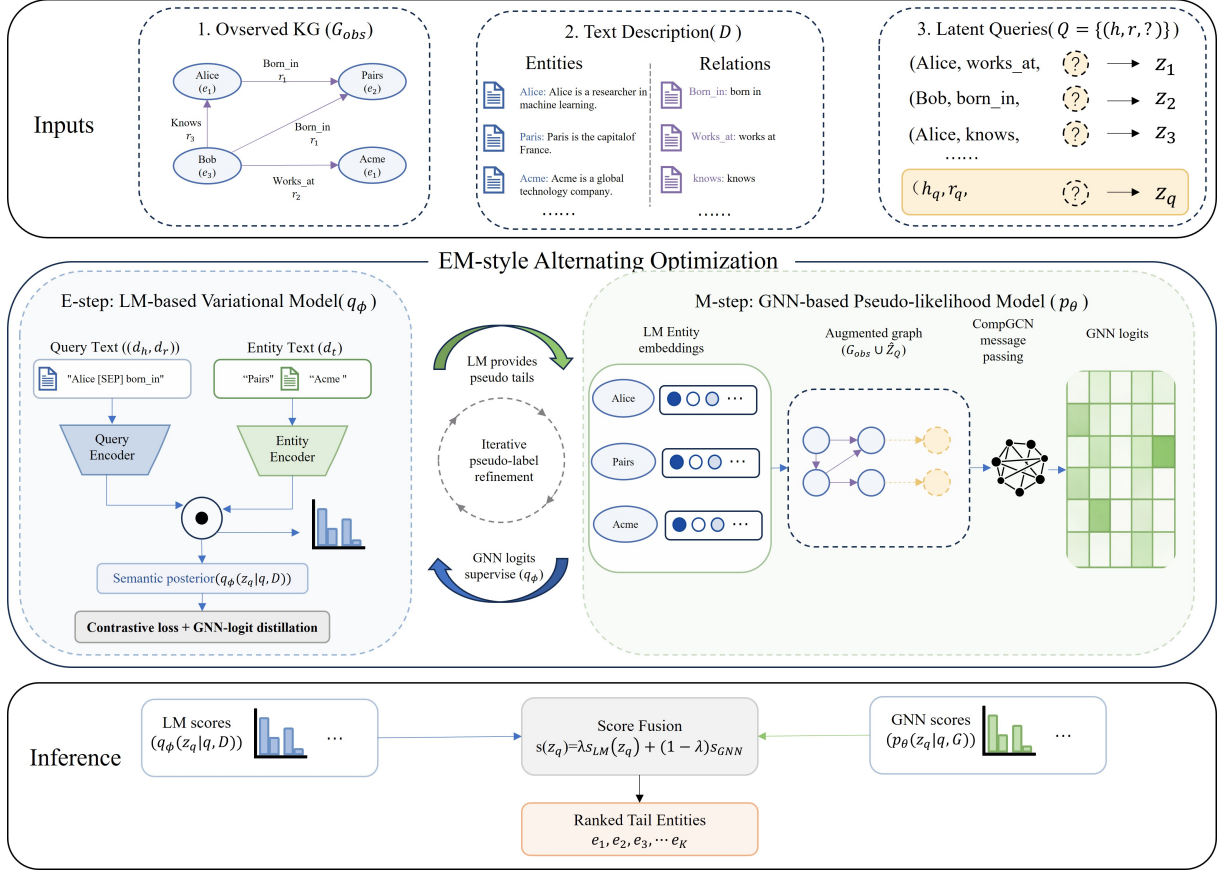


Figure 1: Overall architecture of the proposed pseudo-likelihood variational framework for knowledge graph completion. The LM-based variational model and the GNN-based pseudo-likelihood model exchange semantic and structural evidence through EM-style alternating optimization.

and defines a normalized distribution over the candidate set $\mathcal{C}_q$:

$$q_\phi(z_q = t \mid q, \mathcal{D}) = \frac{\exp(s_\phi(h, r, t)/\tau)}{\sum_{t' \in \mathcal{C}_q} \exp(s_\phi(h, r, t')/\tau)}.$$

The semantic refinement objective contains two parts. First, for observed triples, we follow the supervised contrastive objective of SimKGC (Wang et al., 2022a). Given a mini-batch of triples $\mathcal{B} = \{(h_i, r_i, t_i)\}_{i=1}^B$, each query (h_i, r_i) is matched against its gold tail t_i and the candidate tails appearing in the batch. Let $\mathcal{N}_i$ denote the candidate set consisting of the gold tail and negative entities for the i -th query. The supervised text loss is

$$\mathcal{L}_{\text{sup}}^q = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(s_\phi(h_i, r_i, t_i)/\tau)}{\sum_{t \in \mathcal{N}_i} \exp(s_\phi(h_i, r_i, t)/\tau)}.$$

This objective trains the LM branch to align query texts with correct tail descriptions before incorporating structure-induced supervision from latent queries.

Second, we distill soft structural supervision from the fixed graph model by matching the text distribution to the structure-induced distribution over the same candidate set. The teacher distribution is obtained by temperature-normalizing GNN logits on $\mathcal{C}_q$, and the student distribution is computed from the LM scores on the same candidates:

$$\mathcal{L}_{\text{dist}}^q = \frac{1}{|\mathcal{Q}|} \sum_{q_i \in \mathcal{Q}} \text{KL} \left(\tilde{p}_\theta(\cdot \mid q_i, \mathcal{G}_{\text{obs}}) \parallel q_\phi(\cdot \mid q_i, \mathcal{D}) \right).$$

The approximate E-step objective is

$$\mathcal{L}_E^q = \lambda_{\text{sup}}^q \mathcal{L}_{\text{sup}}^q + \lambda_{\text{dist}}^q \mathcal{L}_{\text{dist}}^q.$$

This update encourages the text model to assign higher probability to structure-consistent latent tails, while preserving the semantic signal from observed triples.

3.4 M-step: Structural Refinement

In the M-step, the text-based model is fixed, and the graph-based model is updated using semantic infor-

mation from the text branch. We first extract LM-derived entity embeddings and project them into the graph representation space with a lightweight adapter:

$$\mathbf{x}_e = \text{MLP}_\psi(\mathbf{v}_e^{\text{LM}}), \quad e \in \mathcal{E},$$

where $\mathbf{x}_e$ is used as the initial node feature of the graph encoder. This allows the graph model to start from semantic entity representations while learning relation-aware structural features.

We instantiate p_θ with CompGCN, which performs relation-aware message passing by composing entity and relation representations. We omit the standard CompGCN update equations here and refer readers to the original CompGCN formulation (Vashishth et al., 2019). After K layers of message passing, the graph encoder outputs final entity and relation representations

$$\mathbf{H}^{(K)} = \{\mathbf{h}_e^{(K)}\}_{e \in \mathcal{E}}, \quad \mathbf{R}^{(K)} = \{\mathbf{h}_r^{(K)}\}_{r \in \mathcal{R}}.$$

Given these representations, the structure-based model scores each candidate tail with a ConvE-style scoring function:

$$s_\theta(h, r, t) = \text{ConvE}_\theta(\mathbf{h}_h^{(K)}, \mathbf{h}_r^{(K)}, \mathbf{h}_t^{(K)}).$$

For a query (h, r) , it produces one-vs-all logits over all candidate entities:

$$\mathbf{s}_\theta(h, r) = [s_\theta(h, r, e)]_{e \in \mathcal{E}}.$$

The M-step incorporates semantic completions inferred by the fixed q_ϕ . For each sampled query $q = (h, r) \in \mathcal{Q}$, the text-based model predicts the top-1 pseudo tail $\tilde{z}_{h,r}^q$. We remove completions that are already observed for the same query, filter out pseudo-edges with confidence below 0.05, and retain the top 90% highest confidence pseudo edges in each refinement round. The resulting pseudo triples are added as typed pseudo edges:

$$G_{\text{aug}} = G_{\text{obs}} \cup \{(h, r, \tilde{z}_{h,r}^q) \mid (h, r) \in \mathcal{Q}\}.$$

CompGCN message passing is performed on G_{aug} , so that LM-inferred pseudo edges directly participate in neighborhood aggregation. Each alternating round consists of one semantic refinement update, one graph-to-text distillation step over sampled latent queries, and one structural training epoch over the pseudo-augmented graph.

The graph model is trained with one-vs-all binary cross-entropy. For an observed query (h, r) ,

let $\mathbf{y}_{h,r} \in \{0, 1\}^{|\mathcal{E}|}$ denote the multi-hot label vector, where $\mathbf{y}_{h,r}[t] = 1$ if $(h, r, t) \in G_{\text{obs}}$. The supervised graph loss is

$$\mathcal{L}_{\text{sup}}^p = \sum_{(h,r) \in \mathcal{T}_{hr}} \sum_{e \in \mathcal{E}} \text{BCE}(\mathbf{y}_{h,r}[e], \sigma(s_\theta(h, r, e))).$$

For pseudo-augmented queries, let $\tilde{\mathbf{y}}_{h,r} \in \{0, 1\}^{|\mathcal{E}|}$ denote the label vector that includes LM-derived pseudo tails. The pseudo-likelihood loss is

$$\mathcal{L}_{\text{PL}}^p = \sum_{(h,r) \in \mathcal{Q}} \sum_{e \in \mathcal{E}} \text{BCE}(\tilde{\mathbf{y}}_{h,r}[e], \sigma(s_\theta(h, r, e))).$$

The M-step objective is

$$\mathcal{L}_M^p = \lambda_{\text{sup}}^p \mathcal{L}_{\text{sup}}^p + \lambda_{\text{PL}}^p \mathcal{L}_{\text{PL}}^p.$$

Through this update, the structure-based model absorbs semantic completion signals from q_ϕ , while refining entity representations through relation-aware message passing over the augmented graph.

3.5 Predictor

After alternating optimization, both the LM-based variational model and the GNN-based pseudo-likelihood model can be used for KGC prediction. For a query $(h, r, ?)$, the LM predictor computes semantic scores

$$s_\phi^{\text{LM}}(h, r, t) = \mathbf{u}_{h,r}^\top \mathbf{v}_t, \quad t \in \mathcal{E},$$

while the GNN predictor computes structure-aware scores

$$s_\theta^{\text{GNN}}(h, r, t) = \text{ConvE}_\theta(\mathbf{h}_h, \mathbf{r}, \mathbf{h}_t).$$

The final predictor combines both sources of evidence:

$$s = (1 - \lambda)N(s_\phi^{\text{LM}}) + \lambda N(s_\theta^{\text{GNN}}).$$

where $\lambda \in [0, 1]$ controls the contribution of structural reasoning and $N(\cdot)$ denotes score normalization. Candidate entities are ranked according to $s(h, r, t)$, and the highest-ranking entities are returned as predicted completions.

3.6 Query-set Construction

The query set $\mathcal{Q}$ defines the support over which latent tail inference is performed. Since the space of unobserved head-relation pairs is both large and heavily skewed, a naive treatment of all missing

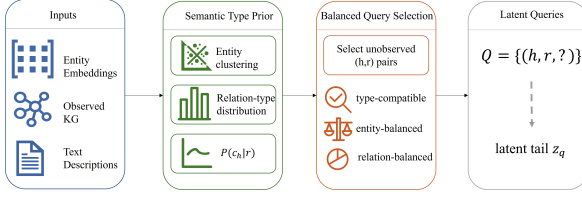


Figure 2: Type-aware balanced query-set construction. LM-derived entity representations induce semantic clusters, from which relation-type priors are estimated. Plausible but unobserved head-relation pairs are then selected with frequency- and degree-aware balancing to form the latent query set $\mathcal{Q}$.

pairs as latent variables would introduce substantial noise and dilute the supervision available to the refinement process. We therefore construct $\mathcal{Q}$ as a restricted set of plausible and distributionally balanced queries, as illustrated in Fig. 2.

The first criterion concerns semantic plausibility. We induce a coarse partition of the entity space from LM-derived entity representations, which are L2-normalized and clustered into semantic regions. For each relation, we estimate an empirical prior over these regions from the observed training triples. A candidate head-relation pair is considered admissible when the semantic region of the head entity is compatible with the relation-specific prior, optionally under relation similarity constraints. This criterion restricts $\mathcal{Q}$ to head-relation pairs that are consistent with patterns already supported by the observed graph.

The second criterion addresses distributional balance. Relations and entities in knowledge graphs are typically long-tailed, and query selection that follows the raw frequency distribution would overemphasize head relations and hub entities. To counteract this effect, candidate queries are reweighted by relation frequency and head-entity degree, and are organized into buckets indexed by relation and entity cluster. Sampling is then carried out under per-entity and per-relation caps, which limits the concentration of $\mathcal{Q}$ on a small number of dominant patterns. In our implementation, the latent-query budget is set to 10% of the augmented training triples for WN18RR and 4.45% for FB15k-237. The detailed cluster numbers, bucket size, per-entity cap, and relation cap are reported in Appendix B.

Taken together, $\mathcal{Q}$ serves as a structured approximation to the missing-edge space, reducing pseudo-label noise and stabilizing the alternating optimization procedure.

4 Experiments

In this section, we evaluate the proposed framework on standard knowledge graph completion benchmarks. Our experiments aim to answer the following research questions:

RQ1: Does the proposed framework improve KGC performance over text-based, graph-based, and hybrid baselines?

RQ2: Can the proposed framework improve completion performance on long-tail relations and low-degree entities?

RQ3: How much does each component of the framework contribute to the final performance?

4.1 Experimental Setup

We evaluate our method on two widely used KGC benchmarks: WN18RR and FB15k-237. WN18RR is derived from WordNet and contains lexical relations between synsets. FB15k-237 is derived from Freebase and contains general-world relational facts. Both datasets remove inverse-relation leakage from their predecessors WN18 and FB15k, making them standard benchmarks for knowledge graph completion. Detailed statistics of the datasets are provided in Appendix A. We evaluate all methods under the standard filtered ranking protocol for knowledge graph completion. We report Mean Reciprocal Rank (MRR) and Hits@ K with $K \in \{1, 3, 10\}$. Hits@ k measures the proportion of test triples whose ground-truth entities are ranked within the top- k predictions, while MRR computes the mean reciprocal rank of the correct entities. Higher metric values indicate better model performance.

We first pretrain the LM and GNN branches independently, and then perform alternating latent-query refinement. The detailed training hyperparameters are provided in Appendix B. Runtime and memory cost are summarized in Appendix D.

4.2 Main Results

Table 1 reports the filtered ranking results on WN18RR and FB15k-237. We compare LQ-KGC with representative structure-based, text-based, and fusion KGC methods. Overall, LQ-KGC outperforms the baselines in nearly all metrics, demonstrating the effectiveness of jointly using textual semantics and graph structure.

On WN18RR, where text-based models are already strong, LQ-KGC further improves the LM branch over SimKGC, especially on Hits@1, sug-

Table 1: Main results on WN18RR and FB15k-237 under the filtered setting. Results for SimKGC and CompGCN are reproduced using the official open-source implementations released by the original authors. Baselines marked with † are taken from SimKGC (Wang et al., 2022a), while all remaining results are reported from the corresponding original papers.

Method	WN18RR				FB15k-237			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
<i>Structure-based methods</i>								
TransE (Bordes et al., 2013)†	24.3	4.3	44.1	53.2	27.9	19.8	37.6	44.1
DistMult (Yang et al., 2014)†	44.4	41.2	47.0	50.4	28.1	19.9	30.1	44.6
ComplEx (Trouillon et al., 2017)†	44.9	40.9	46.9	53.0	27.8	19.4	29.7	45.0
RotatE (Sun et al.)†	47.6	42.8	49.2	57.1	33.8	24.1	37.5	53.3
ConvE (Dettmers et al., 2018)†	43.0	40.0	44.0	52.0	32.5	23.7	35.6	50.1
CompGCN (Vashishth et al., 2019)	47.0	43.7	48.2	53.4	35.1	26.1	38.5	53.3
<i>Text-based methods</i>								
KG-BERT (Yao et al., 2019)†	21.6	4.1	30.2	52.4	—	—	—	42.0
SimKGC (Wang et al., 2022a)	67.1	58.5	73.1	81.7	33.6	24.9	36.2	51.1
GHN (Qiao et al., 2023)	<u>67.8</u>	<u>59.6</u>	71.9	82.1	33.9	25.1	36.4	51.5
<i>Fusion methods</i>								
StAR (Wang et al., 2021a)†	40.1	24.3	49.1	70.9	29.6	20.5	32.2	48.2
CSProm-KG (Chen et al., 2023)	57.5	52.2	59.6	67.8	35.8	26.9	39.3	53.8
Memory-KGC (Li et al., 2026)	68.0	59.8	72.6	81.2	34.0	25.4	37.1	51.4
LQ-KGC _{LM}	68.5	62.1	71.8	80.4	33.9	24.7	37.2	52.4
LQ-KGC _{GNN}	54.1	47.9	56.9	65.5	36.8	27.6	40.2	55.2
LQ-KGC	68.6	62.4	71.8	80.5	38.3	29.1	42.0	56.7

gesting that structure-induced supervision helps sharpen top-ranked predictions. The GNN branch also improves over CompGCN, indicating that LM-derived representations and pseudo completions benefit relation-aware message passing. On FB15k-237, the graph-text interaction is more evident: the LM branch remains close to SimKGC, while the GNN branch gains substantially from semantic initialization and pseudo-edge augmentation. The fused predictor achieves the best overall performance, showing that the two branches capture complementary evidence.

Branch-wise training curves in Appendix G further show that alternating refinement improves ranking metrics across training steps.

4.3 Analysis on Long-tail Completion

We further analyze long-tail completion on WN18RR by grouping test queries according to the degree of the head entity. Figure 3 reports the MRR of different model variants across head-entity degree buckets. The most notable observation is that LQ-KGC_{GNN} consistently improves over vanilla CompGCN in all buckets. The improvement is especially clear for low-degree and medium-degree entities, where the observed graph provides limited neighborhood evidence. This indicates that LM-derived entity representations and pseudo edges effectively compensate for the sparsity of structural

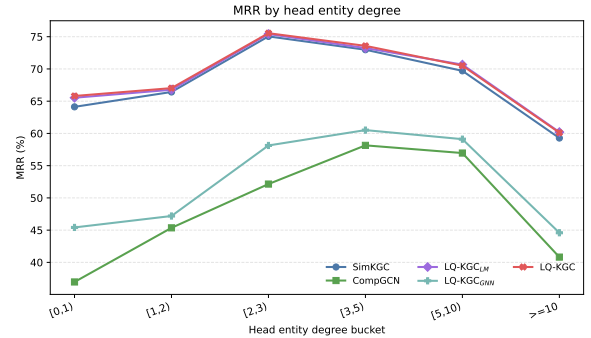


Figure 3: MRR under different head-entity degree buckets on WN18RR. LQ-KGC improves over both SimKGC and CompGCN, especially for low-degree and medium-degree entities.

signals and make relation-aware message passing more informative.

In contrast, the improvement of LQ-KGC_{LM} over SimKGC is relatively small. This is expected because language models already have access to entity and relation descriptions, which provide semantic evidence beyond graph connectivity. For low-frequency or low-degree entities, textual descriptions can partially compensate for the lack of observed neighbors and insufficient training triples. As a result, the LM branch is already strong in sparse regions, leaving less room for additional improvement from structural distillation.

Table 2: Ablation study on WN18RR and FB15k-237.

Variant	Branch	WN18RR				FB15k-237			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Full model	LQ-KGC _{LM}	68.5	<u>62.1</u>	71.8	80.4	33.9	24.7	37.2	52.4
	LQ-KGC _{GNN}	54.1	47.9	56.9	65.5	36.8	27.6	40.2	55.2
	LQ-KGC	68.6	62.4	71.8	<u>80.5</u>	38.3	29.1	42.0	<u>56.7</u>
w/o LM initialization	LQ-KGC _{LM}	68.2	61.7	71.8	80.3	33.8	24.5	37.0	52.4
	LQ-KGC _{GNN}	47.1	44.0	48.0	53.6	35.0	25.9	38.4	53.2
	LQ-KGC	<u>68.5</u>	62.3	71.8	80.4	38.0	28.6	41.6	56.9
w/o LM pseudo edges	LQ-KGC _{LM}	68.4	62.0	71.8	80.5	33.8	24.6	37.2	52.3
	LQ-KGC _{GNN}	52.5	46.3	55.1	64.4	36.3	27.2	39.8	54.9
	LQ-KGC	68.5	62.0	71.8	80.6	<u>38.2</u>	<u>28.9</u>	<u>41.7</u>	56.6
Fusion-only	-	67.9	61.3	71.5	79.9	37.9	28.6	41.5	55.6

4.4 Ablation Study

We perform component-level ablations to evaluate the effectiveness of graph-side semantic integration, pseudo-edge augmentation, and score fusion. We further conduct query-construction ablations to examine the role of the latent-query sampling strategy in latent graph refinement.

4.4.1 Component Ablations

We first evaluate the contribution of each major component. The following variants are considered:

- **w/o LM initialization:** the GNN does not use LM-derived entity embeddings as initial node features.
- **w/o pseudo edges:** LM-inferred pseudo triples are not added to the graph during the M-step.
- **Fusion-only:** independently pretrained LM and GNN scores are combined without latent-query refinement.

Table 2 shows that both LM initialization and pseudo-edge augmentation mainly affect the GNN branch. Removing LM-derived initialization causes a clear drop in GNN performance, especially on WN18RR, where MRR decreases from 54.1 to 47.1. This indicates that textual entity representations provide important initial node features for relation-aware message passing. Removing pseudo edges also weakens the GNN branch on both datasets, suggesting that LM-inferred completions provide useful additional structural evidence during the M-step.

The LM branch is relatively stable across these variants, which is expected because it mainly relies on textual descriptions and supervised contrastive

learning. The fused predictor is also more robust than the standalone GNN branch, but the overall trend still shows that weakening the graph-side semantic input reduces the structural contribution of the framework. Fusion-only is competitive, but the full LQ-KGC further improves MRR by 0.7 on WN18RR and 0.4 on FB15k-237, indicating that the gain does not come from score fusion alone.

We further analyze the contribution of each pseudo-label exchange direction in Appendix H.

4.4.2 Query Construction Ablations

We evaluate the effect of query-set construction by replacing the proposed query sampling strategy with simpler alternatives. For each variant, we report the LM branch, the GNN branch, and the fused predictor.

Random queries samples unobserved head-relation pairs without semantic compatibility filtering or balancing. **Full query construction** combines semantic compatibility and balanced sampling.

Table 3 shows that query construction mainly affects the GNN branch. On WN18RR, full query construction improves GNN MRR to 54.1, compared with 53.3 for random queries. A similar trend is observed on FB15k-237, where the GNN branch improves from 36.2 to 36.8 MRR. These results indicate that balanced query construction provides more useful pseudo edges for graph message passing.

The LM branch and fused predictor are less sensitive to the query construction strategy. This is expected because the LM branch already relies strongly on textual descriptions, and score fusion can partially compensate for noisy query selection. Nevertheless, full query construction gives the strongest GNN branch on both datasets and

Table 3: Ablation on query-set construction.

Variant	Branch	WN18RR				FB15k-237			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Full query construction	LQ-KGC _{LM}	68.5	62.1	71.8	80.4	33.9	24.7	37.2	52.4
	LQ-KGC _{GNN}	54.1	47.9	56.9	65.5	36.8	27.6	40.2	55.2
	LQ-KGC	68.6	62.4	71.8	80.5	38.3	29.1	42.0	56.7
Random queries	LQ-KGC _{LM}	68.4	62.1	71.8	80.4	33.8	24.6	37.0	52.3
	LQ-KGC _{GNN}	53.3	47.0	56.1	65.0	36.2	26.9	39.8	54.4
	LQ-KGC	68.6	62.2	71.8	80.5	38.3	29.2	41.5	56.4

achieves competitive fused performance. This suggests that semantic compatibility improves pseudo-label plausibility, while balanced sampling helps expose the graph model to more informative latent queries rather than concentrating on frequent relations or high-degree entities.

5 Conclusion

In this paper, we proposed LQ-KGC, a latent-query refinement framework that integrates textual semantics and graph structure through alternating pseudo-label refinement. The method constructs plausible latent queries, transfers text-derived pseudo completions to graph reasoning, and uses structure-induced signals to regularize semantic matching. Experiments on WN18RR and FB15k-237 show that LQ-KGC improves MRR and Hits@1 over reproduced text-only and graph-only backbones, with stronger evidence on FB15k-237 and on the GNN branch. The gains are not uniform: on WN18RR, strong text-based baselines remain better on Hits@3 and Hits@10. Our ablations and diagnostics indicate that LQ-KGC is bidirectional in design but asymmetric in effect, with text-to-graph pseudo-edge augmentation providing the main structural benefit.

Limitations

Despite the promising empirical results, our framework still has several limitations. The alternating refinement stage increases the overall training cost, as the text and graph modules must be further optimized after their initial pretraining. Moreover, the construction of latent queries depends on language-model-derived entity embeddings, and thus may degrade when entity descriptions are missing, noisy, or weakly aligned with graph relations. The method may also be less effective on extremely sparse graphs, where CompGCN has too little observed structure to provide reliable structural constraints, or when the query sampler selects

type-incompatible pseudo queries that reinforce prediction errors.

Our empirical scope is confined to two standard transductive KGC benchmarks with textual descriptions. Extending the evaluation to additional text-enhanced datasets and inductive settings requires careful preprocessing, entity-description validation, and leakage control. We also do not yet provide a systematic analysis across relation patterns such as symmetric, compositional, one-to-many, and many-to-one relations. We leave more efficient refinement, more robust latent query construction, full loss-weight sensitivity analysis, and extension to inductive KGC for future work.

Acknowledgements

This work was supported by the National Science and Technology Major Project for Prevention and Control of Emerging and Major Infectious Diseases (Grant No. 2025ZD01900800) and the National Natural Science Foundation of China (Grant Nos. 72371029, 72304269 and 72574215).

References

- 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.
- Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. In *Proceedings of the 29th international conference on computational linguistics*, pages 4005–4017.
- Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. In *Findings of the association for computational linguistics: ACL 2023*, pages 11489–11503.
- Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. Hitter: Hier-

- archical transformers for knowledge graph embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 10395–10407.
- Zhongwu Chen, Long Bai, Zixuan Li, Zhen Huang, Xiaolong Jin, and Yong Dou. 2024. A new pipeline for knowledge graph reasoning enhanced by large language models without fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1381.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687–696.
- Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th international conference on computational linguistics*, pages 1737–1743.
- Feng-Lin Li, Hehong Chen, Guohai Xu, Tian Qiu, Feng Ji, Ji Zhang, and Haiqing Chen. 2020. Alimekg: Domain knowledge graph construction and application in e-commerce. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2581–2588.
- Jiru Li, Yuanyuan Sun, Bo Xu, Dinghao Pan, Ling Luo, and Hongfei Lin. 2026. Memory-kgc: Memory-augmented structural learning for knowledge graph completion. *Information Processing & Management*, 63(7):104891.
- Yu Liu, Yanan Cao, Xixun Lin, Yanmin Shang, Shi Wang, and Shirui Pan. 2025. Enhancing large language model for knowledge graph completion via structure-aware alignment-tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20981–20995.
- Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2395–2405.
- Ananjan Nandi, Navdeep Kaur, Parag Singla, and 1 others. 2024. Dynasemble: dynamic ensembling of textual and structure-based models for knowledge graph completion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 205–216.
- Zile Qiao, Wei Ye, Dingyao Yu, Tong Mo, Weiping Li, and Shikun Zhang. 2023. Improving knowledge graph completion with generative hard negative mining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5866–5878.
- Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2021. [Rnnlogic: Learning logic rules for reasoning on knowledge graphs](#). In *International Conference on Learning Representations (ICLR)*.
- 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 (volume 1: long papers)*, pages 2814–2828.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
- Théo Trouillon, Christopher R Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18(130):1–38.
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.
- Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the web conference 2021*, pages 1737–1748.
- Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022a. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 4281–4294.
- Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021b. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

- Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022b. [Language models as knowledge embeddings](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2291–2297. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kgbert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 535–546.
- Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 233–242.
- Yuan Zhang, Xin Sui, Feng Pan, Kaixian Yu, Keqiao Li, Shubo Tian, Arslan Erdengasileng, Qing Han, Wan-jing Wang, Jianan Wang, and 1 others. 2025. A comprehensive large-scale biomedical knowledge graph for ai-powered data-driven biomedical research. *Nature Machine Intelligence*, 7(4):602–614.
- Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. 2020. [Efficient probabilistic logic reasoning with graph neural networks](#). In *International Conference on Learning Representations (ICLR)*.
- Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations*.

A Dataset Statistics

We present the detailed statistics of the benchmark datasets used in our experiments, including the numbers of entities, relations, and train/validation/test triples in Table 4.

Table 4: Statistics of benchmark knowledge graph datasets.

Dataset	Entities	Relations	Train	Valid	Test	Total
WN18RR	40,943	11	86,835	3,034	3,134	93,003
FB15k-237	14,541	237	272,115	17,535	20,466	310,116

B Training Hyperparameters

For reproducibility, we summarize the main training hyperparameters used in our experiments. We implement our framework in PyTorch and use HuggingFace Transformers for the text encoder. The text branch follows the official SimKGC implementation, and the graph branch is built on the official CompGCN implementation with the same backbone configuration used in our reproduced baseline. The LM and GNN branches are pretrained independently before alternating latent-query refinement. During the alternating stage, semantic refinement updates the LM branch with supervised contrastive learning and distribution alignment, while structural refinement updates the GNN branch on the graph augmented with LM-derived pseudo edges. The remaining hyperparameters introduced by LQ-KGC, including latent-query construction, alternating refinement, and score fusion, are summarized in Table 5.

C Query-level Diagnostics

Table 6 reports the query-level diagnostic under the latent-query budgets used during refinement. It measures whether retained head-relation queries appear in the validation or test splits; it is not a direct KGC evaluation metric.

D Computational Cost

Table 7 reports the training cost measured on a single A100 GPU. The refinement stage adds training overhead after independent LM and GNN pretraining, but it is not used as an iterative procedure at inference time. At inference, LQ-KGC directly computes LM and GNN scores and combines them with the validation-selected fusion weight.

Table 5: Method-specific hyperparameters of LQ-KGC.

Hyperparameter	WN18RR	FB15k-237
Entity Cluster mode	agglomerative	kmeans
Number of clusters	30	200
Latent-query budget	10%	4.45%
Compatibility exponent α	0.5	0.5
Max queries per entity	2	2
Max relation fraction	0.05	0.05
Max bucket size	200	200
Pseudo-edge confidence threshold	0.05	0.05
Retained pseudo-edge ratio	90%	90%
LM pretraining epochs	50	10
GNN pretraining epochs	500	500
Alternating rounds	25	5
Semantic updates per round	1	1
Distillation steps per round	1	1
Structural epochs per round	1	1
λ_{sup}^q	1.0	1.0
λ_{dist}^q	0.2	0.2
λ_{sup}^p	0.6	0.9
λ_{lat}^p	0.4	0.1
Fusion weight λ_{struct}	0.1	0.5

Table 6: Query-level diagnostics under the actual latent-query budgets. Bgt.: budget; Kept: retained queries; V/T: validation/test hit ratio; Cov.: relation coverage.

Data.	Type	Bgt.	Kept	Val.	Test	V/T	Cov.
WN18RR	Rand.	10.00%	15,630	0.37%	0.35%	0.72%	22/22
WN18RR	Ours	10.00%	15,630	2.82%	2.58%	5.39%	22/22
FB-237	Rand.	4.45%	21,805	0.06%	0.15%	0.22%	474/474
FB-237	Ours	4.45%	21,805	4.27%	5.04%	9.08%	462/474

Table 7: Training cost of LQ-KGC on a single A100 GPU.

Dataset	LM pre.	GNN pre.	Refinement	Memory
WN18RR	2.1 h	1.6 h	1.7 h	78.7 GB
FB15k-237	1.6 h	3.1 h	1.1 h	74.7 GB

E Sensitivity to Distillation Weight

As an initial robustness check, we vary the graph-to-text distillation weight λ_{dist}^q on FB15k-237 while keeping the other loss coefficients fixed. Table 8 shows that Fusion MRR remains within 38.18–38.32 across a wide range, indicating that the final predictor is not sensitive to this coefficient. A full joint sensitivity analysis over all loss weights remains future work.

Table 8: Sensitivity to the graph-to-text distillation weight λ_{dist}^q on FB15k-237.

λ_{dist}^q	LM	GNN	Fus.	H@1	H@3	H@10
0.1	33.82	36.42	38.31	29.19	41.88	56.53
0.2	33.90	36.80	38.30	29.10	42.00	56.70
0.3	33.84	36.47	38.27	29.12	41.81	56.62
0.4	33.82	36.44	38.23	29.07	41.81	56.56
0.5	33.83	36.67	38.31	29.16	41.88	56.56
0.6	33.84	36.43	38.25	29.08	41.80	56.59
0.7	33.84	36.51	38.32	29.23	41.78	56.54
0.8	33.86	36.53	38.18	28.96	41.81	56.57
0.9	33.87	36.52	38.28	29.13	41.81	56.60
1.0	33.86	36.62	38.27	29.11	41.89	56.53

F Multi-seed Results

To assess robustness, we report the mean and standard deviation of LQ-KGC variants over three random seeds. The small deviations across runs indicate that the framework is stable under different initializations.

G Training Dynamics

Figure 4 and Figure 5 show the training dynamics of LQ-KGC_{LM} and LQ-KGC_{GNN} across alternating refinement steps. The LM branch improves rapidly in the early stage and then gradually converges, while the GNN branch improves more steadily. This suggests that structure-induced supervision mainly sharpens semantic ranking early, whereas text-derived pseudo completions are absorbed more gradually by graph message passing.

These trends show that alternating refinement is not merely a one-step post-processing effect. Instead, repeated text–graph interaction contributes to progressive metric improvement, with the text branch providing strong semantic matching and the graph branch gradually incorporating additional structural evidence.

H Directional Pseudo-label Exchange Ablations

We further study the effect of pseudo-label exchange directions between the text-based vari-

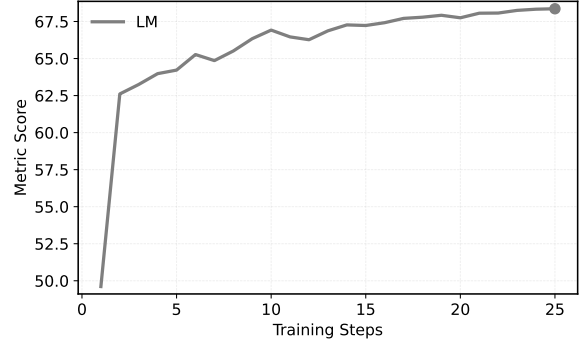


Figure 4: Training dynamics of LQ-KGC_{LM} across alternating refinement steps.

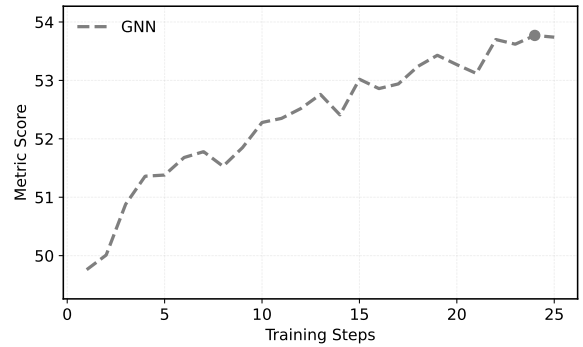


Figure 5: Training dynamics of LQ-KGC_{GNN} across alternating refinement steps.

ational model and the structure-based pseudo-likelihood model. For each variant, we report the LM branch, the GNN branch, and the fused predictor.

Table 10 shows that the Text-to-Graph direction is particularly important for improving the GNN branch. On both datasets, Text-to-Graph only performs closer to the full model than Graph-to-Text only in terms of GNN MRR. This suggests that injecting LM-inferred pseudo completions into the graph is the main source of structural improvement. At the same time, Graph-to-Text distillation still provides useful supervision for the LM branch, although its effect is more modest because the LM branch already has strong textual evidence from entity and relation descriptions. The full model gives the best overall fused MRR, indicating that the two directions contribute asymmetrically but remain complementary: Text-to-Graph mainly strengthens relation-aware message passing, while Graph-to-Text acts as a structure-aware regularizer for semantic ranking.

Table 9: Multi-seed results of LQ-KGC variants on WN18RR and FB15k-237. Mean $\pm$ standard deviation are reported over three runs.

Variant	Branch	WN18RR				FB15k-237			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
LQ-KGC	LM	68.4 $\pm$ 0.2	62.0 $\pm$ 0.2	71.7 $\pm$ 0.1	80.3 $\pm$ 0.3	33.9 $\pm$ 0.0	24.7 $\pm$ 0.0	37.2 $\pm$ 0.0	52.4 $\pm$ 0.1
	GNN	54.1 $\pm$ 0.2	47.9 $\pm$ 0.2	56.9 $\pm$ 0.1	65.4 $\pm$ 0.4	36.6 $\pm$ 0.2	27.4 $\pm$ 0.2	40.0 $\pm$ 0.2	55.0 $\pm$ 0.2
	Full	68.5 $\pm$ 0.2	62.1 $\pm$ 0.3	71.8 $\pm$ 0.1	80.5 $\pm$ 0.3	38.3 $\pm$ 0.1	29.1 $\pm$ 0.1	41.9 $\pm$ 0.1	56.6 $\pm$ 0.1

Table 10: Ablation on directional pseudo-label exchange.

Variant	Branch	WN18RR				FB15k-237			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Full model	LQ-KGC _{LM}	<u>68.5</u>	<u>62.1</u>	<u>71.8</u>	<u>80.4</u>	33.9	24.7	37.2	52.4
	LQ-KGC _{GNN}	54.1	47.9	56.9	65.5	36.8	27.6	40.2	55.2
	LQ-KGC	68.6	62.4	<u>71.8</u>	80.5	38.3	29.1	42.0	56.7
Text-to-Graph only	LQ-KGC _{LM}	68.4	62.1	<u>71.8</u>	80.0	33.9	24.7	37.1	52.5
	LQ-KGC _{GNN}	53.5	47.3	56.1	65.0	36.6	27.5	40.0	54.9
	LQ-KGC	68.5	62.1	71.9	80.3	<u>38.3</u>	<u>29.1</u>	<u>41.9</u>	<u>56.6</u>
Graph-to-Text only	LQ-KGC _{LM}	68.3	61.9	71.6	80.0	33.9	24.7	37.1	52.4
	LQ-KGC _{GNN}	52.3	46.5	54.7	63.2	36.2	26.9	39.8	54.7
	LQ-KGC	68.4	62.1	71.8	80.0	38.2	29.0	41.9	56.5

I Artifacts and Licenses

We use two public KGC benchmarks, WN18RR and FB15k-237. Our implementation is based on PyTorch and HuggingFace Transformers. For baseline reproduction, we refer to the official SimKGC implementation and the official CompGCN implementation. The CompGCN repository is released under the Apache-2.0 license. The public SimKGC repository is used as an academic reference implementation for reproducing the text-based baseline. We will release the preprocessing scripts, query-construction code, and configuration files needed to reproduce the reported LQ-KGC runs.

Use of Generative AI Tools

We acknowledge the use of generative AI tools (e.g., ChatGPT) to assist with language polishing and writing improvements during the preparation of this manuscript.

Specifically, these tools were used to improve grammar, readability, and stylistic consistency of the text. They were not used to generate scientific contributions, experimental designs, model architectures, or conclusions.

All research ideas, theoretical formulations, experiments, and analyses were developed and verified by the authors. The authors take full responsibility for the integrity and correctness of the content of this paper.