Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.087310

ARTICLE

A Dual-Level Structural Context Collaborative Framework for Knowledge Graph Completion

Jing Wang[1], **Tian Xia**[2] **and Hao Li**[1,*]

[1]School of Information Science and Engineering, Yunnan University, Kunming, China
[2]Yunnan Sub-Bureau of Southwest Regional Air Traffic Management Bureau, CAAC, Kunming, China

*Corresponding Author: Hao Li. Email: lihao707@ynu.edu.cn

Received: 14 June 2026; Accepted: 13 August 2026; Published: 15 September 2026

ABSTRACT: Knowledge graphs organize real-world facts as structured triples and have become a fundamental resource for search engines, question answering, recommender systems, and knowledge-enhanced large language models. However, real-world knowledge graphs remain highly incomplete, which limits their downstream reasoning ability. Existing pre-trained language model-based knowledge graph completion methods provide strong textual semantic representations, but they usually model graph structure only as shallow auxiliary features and remain weak in distinguishing structurally similar entities and topology-near negative samples. To address this limitation, this paper proposes a Dual-Level Structural Context Collaborative Framework (DSC^{2}F) for knowledge graph completion. At the instance level, the framework introduces Structural Neighborhood Context (SNC) to inject local neighborhood evidence into the language model input and Relation-Aware Attention (RAA) to condition structural aggregation on the current relation. At the batch level, it constructs topology-aware training batches with biased random walk with restart, so that in-batch negatives are locally related to positive samples and impose stronger structural discrimination pressure. Experiments on WN18RR, FB15k-237, and Wikidata5M show that DSC^{2}F achieves the best mean reciprocal rank and Hits@1 on all three datasets, consistently outperforming strong embedding-based and pre-trained language model-based baselines. Ablation studies and structural configuration analyses further verify that SNC, RAA, and Batch-Level Structural Context provide complementary benefits. These results demonstrate that collaborative modeling of instance-level and batch-level structural context can effectively enhance structure-aware entity representation and improve fine-grained entity prediction.

KEYWORDS: Knowledge graph completion; pre-trained language models; structural context; contrastive learning

1 Introduction

Knowledge graphs (KGs) organize real-world entities and relations as structured triples (h, r, t). They have been widely used in search engines, intelligent question answering, recommender systems, and knowledge-enhanced large language models. Nevertheless, real-world KGs usually suffer from severe incompleteness: many valid entity-relation facts are not explicitly recorded. Knowledge graph completion (KGC), which aims to predict missing entities or relations from observed triples, is therefore a central task in KG research.

Early KGC methods mainly rely on knowledge graph embedding (KGE). Representative models, such as TransE [1], DistMult [2], and ComplEx [3], learn low-dimensional embeddings for entities and relations

and design different scoring functions for entity prediction. These methods capture useful topological regularities, but their representations are generally tied to discrete entity identifiers and lack deep understanding of textual semantics.

With the development of pre-trained language models (PLMs), text-based KGC methods have become an important research direction. KG-BERT [4], StAR [5], and SimKGC [6] exploit entity descriptions and contrastive learning to enhance semantic entity representations. In particular, SimKGC adopts a bi-encoder architecture and enables efficient inference, making it a strong foundation for PLM-based KGC. However, existing PLM-based methods still lack systematic modeling of KG structural context [7,8]. Most of them use structural information as additional shallow features, which makes it difficult for the model to learn discriminative graph regularities, especially when distinguishing structurally similar entities or topology-near negative samples.

This paper argues that the above limitation stems from the lack of unified dual-level structural context modeling. Specifically, two mutually related structural deficiencies have long been treated separately.

First, existing methods lack instance-level structural context. Most PLM-based methods encode a triple (h, r, t) only according to textual descriptions and do not explicitly perceive the local topological environment. Consequently, the model cannot adequately characterize the dependency between an entity neighborhood and the semantics of the current relation, and it has to complete entity matching mainly through textual semantics. However, entity connections in a KG are not determined only by text: local topological patterns often provide key discriminative evidence. Different relations usually correspond to different neighborhood aggregation patterns, and the same entity may activate different structural associations under different relation conditions. Without instance-level structural context, the model can hardly establish stable structural-semantic alignment, is prone to confusion among structurally similar entities, and shows weaker generalization on complex relation patterns.

Second, existing methods lack batch-level structural context. Mainstream contrastive learning frameworks usually adopt random negative sampling, where negative samples are often independent of positive samples in the KG. The model can therefore distinguish positives from negatives using shallow textual differences without truly learning structural boundaries. Although this training strategy improves semantic matching, it is insufficient for strengthening structural discrimination and easily leads to incorrect predictions for topologically close entities. Random negative sampling also fails to construct hard negatives with realistic topological interference, limiting the model's ability to learn fine-grained structural differences.

To address these problems, this paper proposes a dual-level structural context collaborative framework for KGC, named DSC^{2}F, which uniformly models KG structure at the instance and batch levels. At the instance level, the framework builds instance-level structural context (ISC) around each triple through structural neighborhood context (SNC) and relation-aware attention (RAA). SNC explicitly aggregates entity neighborhood information to form local topological representations, while RAA injects relation semantics into the Transformer attention layer and relation-conditionally modulates the neighborhood aggregation process, thereby improving consistency between local structure and the current relation. At the batch level, the framework introduces batch-level structural context (BSC) and constructs topology-aware batches through biased random walk with restart (BRWR). Topologically constrained subgraph sampling keeps in-batch negatives locally close to positive samples and creates hard negatives with realistic structural interference. Under this training mechanism, the model more fully uses instance-level structural context to distinguish candidates that are textually similar but structurally different.

Based on these designs, DSC^{2}F forms a complete dual-level structural collaboration loop: instance-level structural context establishes local structural semantics inside each triple, while batch-level structural

context continuously imposes structural discrimination pressure during training. The two levels jointly drive the model to learn structure-aware representations.

The main contributions of this paper are summarized as follows:

- We propose a dual-level structural context collaborative framework for KGC, which uniformly models KG structural information at both the instance and batch levels. At the instance level, the framework incorporates structural neighborhood context (SNC) into PLM encoding and uses relation-aware attention (RAA) to jointly model local neighborhood structure and relation-conditioned semantics. At the batch level, the framework constructs topology-aware training batches through biased random walk with restart, generating structural hard negatives with realistic topological proximity and thereby enabling collaborative optimization between local structural encoding and structure-discriminative training.
- Experiments on multiple public KGC datasets show that DSC^{2}F effectively improves the ability of PLM-based models to handle structurally similar entities, topology-near negative samples, and complex relation patterns, achieving stable performance gains on entity prediction. Ablation studies and structural configuration analyses further verify the complementary effects of SNC, RAA, and batch-level structural context, demonstrating that dual-level structural context collaboration enhances structural awareness at both the encoding and training levels.

2 Related Work

2.1 Knowledge Graph Completion

Knowledge graph completion (KGC) aims to infer missing triples in a knowledge graph. Traditional methods are mainly based on knowledge graph embedding (KGE), such as TransE [1] and RotatE [9]. These methods define specific distance functions or scoring functions to model entities and relations in a low-dimensional continuous vector space. Subsequently, graph neural network (GNN) based methods, such as R-GCN [10] and CompGCN [11], were proposed to better capture graph topological information through message passing.

In recent years, with the rise of pre-trained language models (PLMs), text-based KGC methods have shown strong generalization ability. KG-BERT [4] first treats triples as textual sequences and feeds them into BERT for binary classification. They must traverse the entire entity set during inference, resulting in prohibitively high computational cost. To alleviate this problem, bi-encoder architectures have been introduced, including StAR [5] and SimKGC [6]. In particular, SimKGC significantly improves both inference efficiency and entity prediction performance through a bi-encoder architecture and in-batch negatives. To further enhance the structural awareness of PLM-based KGC models, recent methods such as PEMLM [12] and ProgKGC [13] attempt to construct entity representations that combine semantics and structure through pre-encoded semantic representations, structural embedding fusion, and progressive structural enhancement. More recently, large language models (LLMs) have also been explored for KGC. KG-LLM [14] formulates triples as text sequences and uses entity and relation descriptions as prompts for LLM-based prediction, while KICGPT [15] combines an LLM with a triple-based retriever and encodes structural knowledge as in-context demonstrations to improve completion performance without additional fine-tuning.

Recent studies have further explored deeper integration between textual representations and KG structure. RAA-KGC [16] constructs relation-aware anchor entities from the head-entity neighborhood and uses them to refine query representations. SLiNT [17] injects neighborhood-derived structural context into a frozen language model and combines this mechanism with dynamic hard contrastive learning. PEKGC [18] jointly models triple-level and path-level evidence and introduces a neighbor selector to filter

adjacent structural information. From the LLM perspective, SAT [19] aligns graph embeddings with the language representation space through hierarchical knowledge alignment and structural instruction tuning. These advances demonstrate a growing trend toward neighborhood enhancement, path-based reasoning, structure-aware contrastive learning, and graph–language alignment. However, they mainly address individual forms of structural enhancement, whereas DSC^{2}F collaboratively models relation-conditioned neighborhood context at the instance level and topology-aware hard-negative construction at the batch level.

2.2 Prompt Tuning for KGC

Prompt tuning was originally proposed to more fully elicit the generalization potential of pre-trained language models (PLMs) or large language models in few-shot and even zero-shot scenarios [20–22]. Such methods usually introduce discrete prompts or continuous learnable prompts into the input, guiding the model to quickly adapt to specific downstream tasks while avoiding full parameter fine-tuning.

In the field of KGC, prompt mechanisms have also been used to promote the fusion of structural information and textual semantics. For example, CSProm-KG [23] generates conditional soft prompts from entity and relation representations to inject structural knowledge into PLMs. TAGREAL [24] automatically constructs query prompts for open KGC and combines external text retrieval to enhance the knowledge probing ability of PLMs. These studies show that prompt mechanisms can provide an effective task adaptation approach for PLM-based KGC. However, these prompt-based methods still mainly transform entities, relations, or retrieved texts into prompt tokens in a one-dimensional sequence, and their characterization of the structural context around triples remains insufficient. In particular, the local neighborhood of an entity often contains relation-dependent clues, and the same entity may provide different structural evidence under different relations. Therefore, how to better utilize neighborhood structure and relation-conditioned contextual information to improve knowledge graph completion remains an urgent problem to solve.

2.3 Random Walks and Structured Negative Sampling

Random walk is a classical strategy for capturing node sequences and local topology in graph data analysis and representation learning. From early methods such as DeepWalk [25] and node2vec [26] to RDF2Vec [27] for RDF knowledge graphs, random-walk sequences have been widely used for learning node and entity representations. Meanwhile, random walk with restart (RWR) [28] periodically returns to the starting node to strengthen local neighborhood exploration, providing an important basis for local structural modeling and subgraph sampling. In KGC, constructing high-quality hard negatives from graph topology to improve contrastive learning has long been an important research topic.

As this research direction has developed, negative sampling methods have gradually evolved from random replacement to more structure-aware and model-aware hard negative construction. KBGAN [29] generates more challenging negative triples through adversarial learning. NSCaching [30] efficiently maintains high-quality negatives through a cache mechanism. Structure Aware Negative Sampling [31] further uses KG structural information to constrain negative sample selection.

Different from the above methods, which mainly focus on adversarial generation, cache-based filtering, or direct enhancement of representation learning objectives, this paper introduces biased random walk with restart (BRWR) for structural hard negative mining to construct batch-level structural context. In this way, positive and negative samples participating in contrastive learning within the same batch possess local topological proximity, thereby imposing realistic topological discrimination pressure on the language model from the perspective of the optimization objective.

3 Method

This paper proposes a dual-level structural context collaborative framework for knowledge graph completion. By collaboratively modeling instance-level structural context and batch-level structural context, the framework effectively injects local structural dependencies from the KG into a pre-trained language model (PLM). The overall architecture is shown in Fig. 1. The first layer is instance-level structural context, which injects local structural semantics for entities h, t through structural neighborhood context, enables the model to perceive the neighborhood topological environment during encoding, and introduces relation-aware attention to help the PLM aggregate local structural information. The second layer is batch-level structural context, whose core goal is to construct hard negatives with structural ambiguity during training, thereby encouraging the model to learn more fine-grained structural discrimination ability.

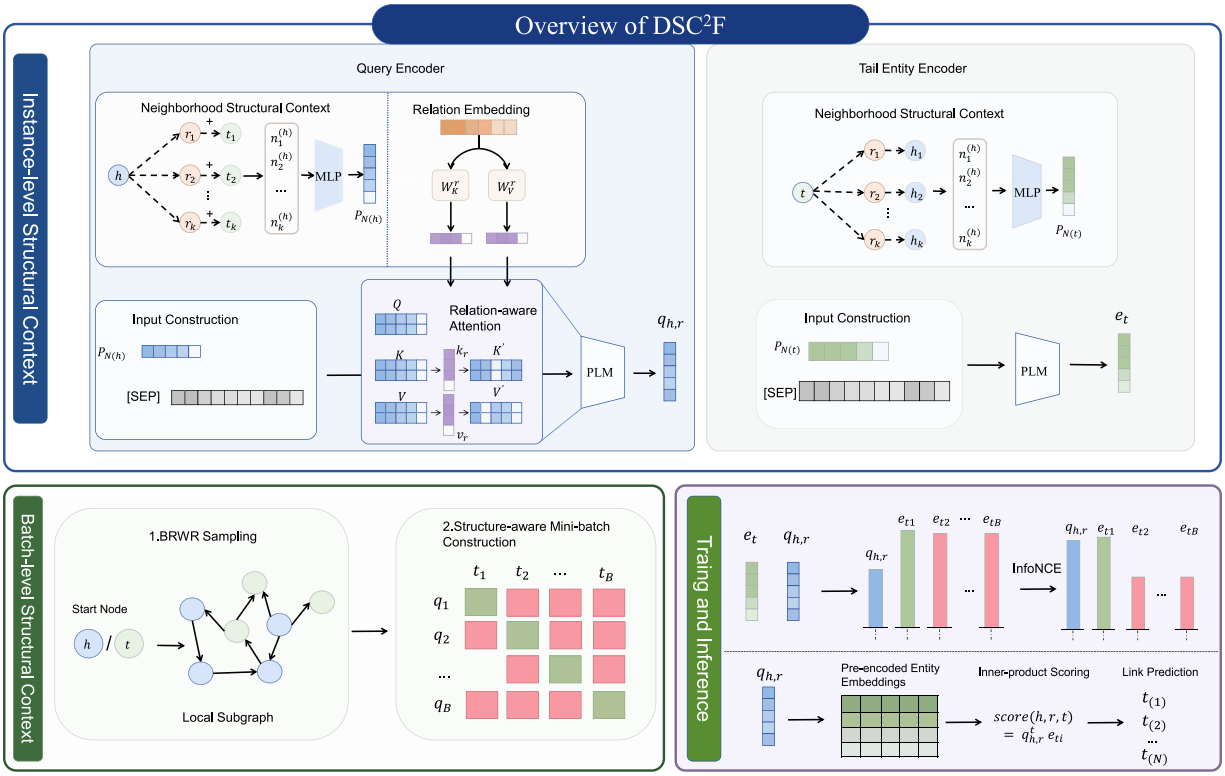

Figure 1: Overview of the proposed DSC²F framework.

3.1 Instance-Level Structural Context

Unlike conventional bi-encoders that rely only on entity textual descriptions for representation learning, this paper argues that candidate entity discrimination in KGC depends not only on textual semantic similarity but also on the local structural evidence required by the current query. Specifically, given a query $(h, r, ?)$, the model needs to determine whether a candidate tail entity is located in a structural environment compatible with the query. To this end, this paper proposes instance-level structural context and designs structural neighborhood context (SNC) and relation-aware attention (RAA), which jointly construct local structural semantics for each instance at the input layer and the attention computation layer.

3.1.1 Structural Neighborhood Context

Structural neighborhood context explicitly encodes the local neighborhood structure around an entity, so that the model can perceive instance-level local structural information at the input stage.

Let the KG be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the entity set, relation set, and triple set, respectively. We augment the relation vocabulary with inverse relations, denoted by $\widetilde{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1}$. The model maintains a shared entity embedding matrix $\mathbf{E} \in \mathbb{R}^{(|\mathcal{E}|+1)\times d}$ and a relation embedding matrix $\mathbf{R} \in \mathbb{R}^{(|\widetilde{\mathcal{R}}|+1)\times d}$. The additional row in each matrix is reserved for the entity and relation padding tokens used when an entity has fewer than σ neighbors.

For any entity x, its normalized one-hop neighborhood is defined as

$$\mathcal{N}(x) = \{(r, u) \mid (x, r, u) \in \mathcal{T}\} \cup \left\{(r^{-1}, u) \mid (u, r, x) \in \mathcal{T}\right\}. \tag{1}$$

Thus, outgoing edges are retained in their original direction, whereas each incoming edge (u, r, x) is rewritten as (x, r^{-1}, u) before neighborhood encoding. Taking the head entity h as an example, let $(r_i, h_i) \in \mathcal{N}(h)$ denote its i-th normalized neighbor pair. Its structural representation is defined as

$$\mathbf{n}_i^{(h)} = \mathbf{e}_{h_i} + \mathbf{r}_{r_i}, \tag{2}$$

where $\mathbf{e}_{h_i}, \mathbf{r}_{r_i} \in \mathbb{R}^d$ are the embedding vectors retrieved from $\mathbf{E}$ and $\mathbf{R}$, respectively. Similarly, the neighbor representation of a candidate-side entity is

$$\mathbf{n}_j^{(t)} = \mathbf{e}_{t_j} + \mathbf{r}_{r_j}. \tag{3}$$

A shared MLP is then used to nonlinearly map neighborhood structural information:

$$\mathbf{P}_{\mathcal{N}(x)} = \mathrm{MLP}\left(\left[\mathbf{n}_1^{(x)}; \ldots; \mathbf{n}_\sigma^{(x)}\right]\right) \in \mathbb{R}^{\sigma\times d}, \quad x \in \{h, t\}. \tag{4}$$

Here, $\mathbf{P}_{\mathcal{N}(x)}$ is a matrix composed of multiple neighborhood structural vectors, and each row corresponds to a neighbor structural representation after MLP mapping. To reduce sensitivity to arbitrary neighbor ordering, all injected neighborhood structural prompts are assigned the same reserved absolute position index of 511, and their position indices are concatenated with those of the text tokens before being fed into the Transformer. This design distinguishes the neighborhood prompt block from the text block while treating the neighbors as a unified positional block, so their internal relative order is not indicated by distinct absolute position embeddings.

Finally, the neighborhood prompt matrix is injected into the input layer of the PLM as a structural prefix:

$$\mathbf{H}_{\mathrm{aug}}^{(0)}(x) = \left[\mathbf{P}_{\mathcal{N}(x)}; \mathbf{H}_x^{(0)}\right] \in \mathbb{R}^{(\sigma+n)\times d}, \quad x \in \{h, t\}. \tag{5}$$

Here, $\mathbf{H}_x^{(0)}$ denotes the original textual embedding sequence of the corresponding encoder, n denotes the length of the original input text sequence, and σ denotes the fixed number of injected neighborhood prompt vectors. Through this mechanism, the model can perceive neighborhood information around entities at the instance level and establish local structural context for a single triple.

3.1.2 Relation-Aware Attention

Structural neighborhood context provides the model with the local topological environment of entities. However, after structural prompts are introduced, the model needs to process interactions among entity text, relation text, and neighborhood structural information. In this process, neighborhood structural information may interfere with the modeling of the current relation semantics and cause structural aggregation to deviate from the current relation context.

Based on this observation, this paper proposes relation-aware attention. It directly injects relation conditions into the query-side Transformer attention layer, providing relation-conditioned constraints for structural information aggregation. This alleviates the inconsistency between neighborhood structural information and the current relation semantics and enhances the model's ability to capture relation-aware instance-level structural context.

For the relation embedding vector $\mathbf{r}_r \in \mathbb{R}^d$, it is first projected into the attention space:

$$\mathbf{k}_r = \mathbf{r}_r \mathbf{W}_K^r, \quad \mathbf{v}_r = \mathbf{r}_r \mathbf{W}_V^r. \tag{6}$$

Let the hidden state of the l-th Transformer layer be $\mathbf{H} \in \mathbb{R}^{(\sigma+n)\times d}$. The Key and Value matrices in standard attention are

$$\mathbf{K} = \mathbf{H}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}\mathbf{W}_V. \tag{7}$$

The relation-conditioned vectors are concatenated to the ends of Key and Value:

$$\mathbf{K}'; = \begin{bmatrix} \mathbf{K} \\ \mathbf{k}_r \end{bmatrix}, \quad \mathbf{V}' = \begin{bmatrix} \mathbf{V} \\ \mathbf{v}_r \end{bmatrix}. \tag{8}$$

The corresponding attention computation is

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}', \mathbf{V}') = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}'^{\top}}{\sqrt{d_k}}\right)\mathbf{V}'. \tag{9}$$

Under this mechanism, the relation vector acts as a conditional modulation signal for structural information aggregation and adjusts the attention distribution. This relation-conditioned signal dynamically changes the relative weights of textual information and structural prompts during attention computation, enabling the model to consider the current relation semantics more strongly when aggregating local structural information. Therefore, RAA establishes a relation-conditioned structural modulation mechanism, allowing the model to better coordinate local structural information and current relation semantics inside a single triple, thereby assisting target tail entity prediction.

3.2 Batch-Level Structural Context

Instance-level structural context mainly models local structural semantics under relation conditions for a single triple. If training batches are constructed only by random sampling from global triples, in-batch negatives often lack topological relevance. The model may then rely on shallow semantic differences for discrimination and fail to learn structural discriminative features.

To address this issue, this paper further constructs batch-level structural context. The method reconstructs the sampling distribution through local graph topology constraints, making candidate tail entities in the same batch more strongly neighborhood-related in structure and increasing the difficulty of structural discrimination in contrastive learning.

Specifically, for each center triple (h, r, t) in the training set, biased random walk with restart (BRWR) is first used to construct a local subgraph $\mathcal{G}_{\text{sub}}$ in the KG. The starting node $s \in \{h, t\}$ is selected according to an inverse-degree normalized probability:

$$P(s = u) = \frac{|\Gamma(u)|^{-1}}{|\Gamma(h)|^{-1} + |\Gamma(t)|^{-1}}, \quad u \in \{h, t\}, \tag{10}$$

where $\Gamma(u)$ denotes the node-level neighbor set of node u.

Starting from s, a random walk is then performed. At each step, the walk returns to s with probability p_r and transfers among the neighbor set with probability $1 - p_r$. To reduce the dominance of highly connected nodes during sampling, a neighbor-normalized weight is introduced:

$$\alpha(v) = \frac{1}{|\Gamma(v)|}, \tag{11}$$

and the state transition probability is defined as

$$P_{\text{trans}}(v|u) = \frac{\alpha(v)}{\sum_{v_j \in \Gamma(u)} \alpha(v_j)}. \tag{12}$$

The triples visited by the random walk are then used to construct the corresponding local subgraph $\mathcal{G}_{\text{sub}}$.

Based on this subgraph, a subgraph-driven batch construction strategy samples $|\mathcal{B}|/2$ triples from $\mathcal{G}_{\text{sub}}$ to form the training batch. To avoid excessive repetition of high-frequency entities in local sampling, a frequency constraint mechanism based on local statistics is introduced. Specifically, for the current subgraph $\mathcal{G}_{\text{sub}}$, let $\mathcal{E}_{\text{sub}}$ be the unique entity set contained in it. The average entity-level sampling capacity is defined as

$$\gamma = \left\lceil \frac{M}{|\mathcal{E}_{\text{sub}}|} \right\rceil. \tag{13}$$

Here, M denotes the number of triples contained in the local subgraph $\mathcal{G}_{\text{sub}}$. During batch construction, if an entity has appeared in the current batch up to the upper bound γ, subsequent candidate triples containing this entity are skipped until $|\mathcal{B}|/2$ triples have been sampled.

Finally, to support bidirectional prediction, the sampled triples and their inverse-relation triples (t, r^{-1}, h) are both added to the batch $\mathcal{B}$:

$$\mathcal{B} = \{(h_i, r_i, t_i)\}_{i=1}^{|\mathcal{B}|}. \tag{14}$$

This strategy jointly optimizes the batch distribution through local structural constraints and frequency control, thereby providing more discriminative training signals for structure-aware representation learning.

3.3 Training and Inference

Based on the above dual-level structural context modeling, this paper constructs a unified structure-aware contrastive learning framework. Here, $\mathbf{E}$ and $\mathbf{R}$ retain their definitions in Section 3.1.1 as trainable lookup matrices for constructing neighborhood prompts, whereas $\mathbf{e}_t$ denotes the final candidate representation produced by the encoder.

Training stage For a given training triple (h, r, t), the query-side encoder combines SNC and RAA to produce the relation-conditioned query vector $\mathbf{q}_{h,r}$, whereas the candidate-side (tail entity) encoder uses SNC alone to produce the relation-independent tail entity vector $\mathbf{e}_t$. Before constructing the contrastive objective, known-positive masking is applied to the in-batch candidates. Specifically, the filtered negative set is defined as

$$\mathcal{N}_{\mathcal{B}}(h, r) = \{t_i \mid (h_i, r_i, t_i) \in \mathcal{B},\ t_i \neq t,\ (h, r, t_i) \notin \mathcal{T}_{\text{train}}\}, \tag{15}$$

where candidate tails that form observed training triples with the current head–relation query are excluded. Contrastive learning is then performed with the InfoNCE loss:

$$\mathcal{L}_{(h,r,t)} = -\log \frac{\exp(\mathbf{q}_{h,r}^{\top}\mathbf{e}_t/\tau)}{\exp(\mathbf{q}_{h,r}^{\top}\mathbf{e}_t/\tau) + \sum_{t_i \in \mathcal{N}_{\mathcal{B}}(h,r)} \exp\left(\mathbf{q}_{h,r}^{\top}\mathbf{e}_{t_i}/\tau\right)}. \tag{16}$$

Here, $\mathcal{B}$ denotes the current training batch, and τ is the temperature coefficient. Known-positive masking prevents observed alternative answers from being incorrectly penalized as negatives. Through this training mechanism, instance-level structural context provides local structural supplementary evidence for the model, while batch-level structural context introduces hard constraints from the local topology. Together, they promote structure-aware representation learning.

Inference stage Although **E** and **R** are updated during training, all parameters are frozen after training, and the relation-independent candidate entity vectors $\mathbf{e}_{t_i}$ produced by the candidate-side encoder are computed and cached. For the entity prediction task $(h, r, ?)$, the query-side encoder combines SNC and RAA to generate $\mathbf{q}_{h,r}$. It then computes the inner product between this query vector and all pre-stored candidate entity vectors $\mathbf{e}_{t_i}$ in the KG to estimate the confidence score of each candidate triple:

$$\text{score}(h, r, t_i) = \mathbf{q}_{h,r}^{\top}\mathbf{e}_{t_i}. \tag{17}$$

Finally, all candidate entities are ranked in descending order according to their scores to complete entity prediction.

3.4 Computational Complexity and Memory Overhead

We analyze the efficiency of DSC^{2}F according to its implementation. Let B denote the batch size, n the text sequence length, σ the number of sampled neighbors, p the number of relation-conditioned Key/Value prefix entries per Transformer layer, d the hidden dimension, and L the number of Transformer layers. The augmented input length is denoted by $N = n + \sigma$. In the following analysis, constant factors arising from the fixed number of encoder calls are omitted.

3.4.1 Instance-Level Structural Context

Structural Neighborhood Context. SNC performs entity and relation lookup, vector addition, and a two-layer MLP projection. Their respective costs are $O(B\sigma d)$, $O(B\sigma d)$, and $O(B\sigma d^2)$. More importantly, SNC increases the Transformer input length from n to N. Compared with the text-only Transformer cost $O(BL(nd^2 + n^2 d))$, the additional encoding cost introduced by the longer sequence is

$$O\left(BL\left[\sigma d^2 + \left(2n\sigma + \sigma^2\right) d\right]\right). \tag{18}$$

Relation-Aware Attention. The implementation generalizes the relation-conditioned vectors defined in Section 3.1.2 to p Key vectors and p Value vectors at each Transformer layer. These additional vectors extend the Key/Value sequence length without introducing additional Query tokens. Generating these vectors requires $O(BpLd^2)$ computation, while the additional attention interaction requires $O(BLNpd)$ computation and $O(BLpd)$ temporary Key/Value memory. The resulting attention matrix has size $N \times (N + p)$.

Combining SNC and RAA, the instance-level encoder has the following principal time complexity:

$$O\Big(BL\left[Nd^2 + N(N+p)d\right] + B\sigma d^2 + BpLd^2\Big). \tag{19}$$

3.4.2 Batch-Level Structural Context

BRWR is executed as an offline preprocessing procedure rather than inside the model forward pass. Constructing the graph and inverse-degree transition weights requires linear preprocessing in the graph size and $O(|\mathcal{E}| + |\mathcal{T}|)$ graph storage. For C center triples, at most I walk iterations and K steps per iteration result in $O(CIK\bar{\delta})$ time in the current implementation, where $\bar{\delta}$ denotes the average cost of scanning a node's neighbor list during biased selection. The resulting dictionary requires $O(CS)$ storage for subgraphs of fixed size S. During training, the stored subgraphs are read to form batches, so BRWR introduces no online random-walk computation, although it incurs linear batch-construction and data-loading cost.

Because BRWR changes the composition rather than the size of a batch, the in-batch contrastive score matrix retains the same $O(B^2d)$ computation and $O(B^2)$ memory as the underlying bi-encoder framework. Thus, batch-level structural context mainly transfers its cost to offline preprocessing and storage.

3.4.3 Additional Parameters and Training Memory

Table 1 summarizes the additional trainable parameters. Here, $\widetilde{\mathcal{R}}$ denotes the relation vocabulary containing both original and inverse relations. The RAA encoder follows the implemented two-layer projection from a relation embedding to the Key/Value vectors of all L layers.

Table 1: Additional trainable parameters introduced by DSC^{2}F.

Component	Parameter Count	FP32 Parameter Storage
Entity lookup matrix	$(\lvert\mathcal{E}\rvert + 1)d$	120 MiB/43 MiB/13.15 GiB$^{\leqq}$
Relation lookup matrix	$(\lvert\widetilde{\mathcal{R}}\rvert + 1)d$	Less than 5 MiB
Neighborhood MLP	$4d^2 + 3d$	Approximately 9.0 MiB$^{\geqq}$
RAA encoder	$(\lvert\widetilde{\mathcal{R}}\rvert + 1)d + (1 + 2pL)d^2 + (1 + 2pL)d$	Depends on p and L

Note: $^{\leqq}$ Values correspond to WN18RR, FB15k-237, and Wikidata5M, respectively, with $d = 768$. $^{\geqq}$ Computed for BERT-base with $d = 768$.

The entity lookup matrix is the dominant additional parameter storage on Wikidata5M. For activations, the attention-map component increases from $O(BLn^2)$ to $O(BLN(N + p))$, the hidden-state component increases from $O(BLnd)$ to $O(BLNd)$, and RAA additionally stores $O(BLpd)$ temporary Key/Value entries.

3.4.4 Inference Efficiency and Comparison with SimKGC

After training, candidate entity representations are computed and cached. BRWR is not used during inference. For each query, DSC^{2}F performs structure-aware query encoding followed by inner-product scoring against the candidate matrix. Table 2 compares the principal complexity with SimKGC [6].

Table 2: Principal computational complexity of SimKGC and DSC^{2}F.

Method	Training	Online Inference Per Query
SimKGC	$O(BL(nd^2 + n^2d) + B^2d)$	$O(L(nd^2 + n^2d) + \lvert\mathcal{E}\rvert d)$
DSC^{2}F	$O(BL[Nd^2 + N(N+p)d] + B\sigma d^2 + BpLd^2 + B^2d)$	$O(L[Nd^2 + N(N+p)d] + \sigma d^2 + pLd^2 + \lvert\mathcal{E}\rvert d)$

DSC^{2}F therefore introduces bounded query-encoding overhead controlled by σ and p, while retaining the same $O(|\mathcal{E}|d)$ candidate-scoring complexity and $O(|\mathcal{E}|d)$ candidate-cache requirement as a standard bi-encoder. Its main scalability costs are the augmented training sequence and the entity lookup matrix, whereas BRWR affects only offline preprocessing and subgraph storage.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

To systematically evaluate the effectiveness of the proposed dual-level structural context collaborative framework for KGC, this paper conducts experiments on three widely used public benchmark datasets: WN18RR [32], FB15k-237 [33], and Wikidata5M [34]. These datasets cover lexical semantic KGs, general entity-relation KGs, and large-scale open-domain KGs, respectively, and therefore evaluate entity prediction ability under different structural characteristics, relation complexities, and graph scales. Table 3 reports the numbers of entities, relations, and triples in the training, validation, and test sets.

Table 3: Summary statistics of benchmark datasets.

Dataset	Entities	Relations	Training	Validation	Test
WN18RR	40,943	11	86,835	3034	3134
FB15k-237	14,541	237	272,115	17,535	20,466
Wikidata5M	4,594,485	822	20,614,279	5163	5163

WN18RR is derived from WordNet and mainly contains English lexical items and semantic relations. It removes inverse-relation leakage from WN18, making it more suitable for evaluating genuine reasoning over semantic relations and local topology. FB15k-237 is derived from Freebase and covers real-world entities with multiple relation types. Compared with WN18RR, FB15k-237 contains more relation types and more complex entity interaction patterns, which makes it useful for testing structural discrimination in multi-relation scenarios. Wikidata5M is constructed from Wikidata and Wikipedia page information and contains large-scale entities, relations, and textual descriptions. Compared with WN18RR and FB15k-237, it is much larger in both entity count and training triples, and thus further evaluates scalability, representation learning, and structure-aware training on open-domain KGs.

4.1.2 Evaluation Metrics

KGC is usually formulated as a ranking task. Given a query, a model scores all candidate tail entities and ranks candidate triples accordingly. A higher position of the true tail entity indicates stronger entity prediction ability. Following the standard evaluation protocol for KGC, this paper reports MRR, Hits@1, Hits@3, and Hits@10.

MRR measures the average reciprocal rank of the true entity and is defined as

$$\mathrm{MRR} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{\mathrm{rank}_q}, \tag{20}$$

where $\mathcal{Q}$ is the test query set and rank_q is the rank of the true entity among all candidate entities for query q. A higher MRR indicates that true answers are ranked higher overall.

Hits@k measures whether the true entity appears in the top k ranked candidates:

$$\mathrm{Hits@}k = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}(\mathrm{rank}_q \leq k), \tag{21}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Hits@1, Hits@3, and Hits@10 measure whether the correct entity appears within the top 1, 3, and 10 ranked candidates, respectively.

4.1.3 Implementation Details and Baselines

We use BERT-base-uncased as the PLM backbone for both the query-side encoder and the candidate-side encoder, with a maximum text length of 50 tokens. All experiments are conducted on three RTX PRO 6000 GPUs with PyTorch 2.7.0. The learnable temperature coefficient in the InfoNCE loss is initialized to 0.05, the learning rate is set to 1×10^{-5}, and the restart probability of BRWR is set to 0.04. The number of relation-conditioned Key/Value vectors per Transformer layer is set to $p = 4$ for all datasets. The training batch size for WN18RR and Wikidata5M is selected from $\{512, 1024, 1536, 2048\}$, while the batch size for FB15k-237 is selected from $\{1024, 2048, 3072, 4096\}$. The number of sampled neighbors is selected from $\{0, 4, 8, 16, 32, 64\}$. WN18RR is trained for 50 epochs with a total time of about 4 h; FB15k-237 is trained for 30 epochs with a total time of about 5 h; Wikidata5M is trained for 2 epochs with a total time of about 25 h. For statistical significance testing, DSC^{2}F and SimKGC are evaluated in three paired runs using identical random seeds and dataset splits. We conduct two-tailed paired t-tests with two degrees of freedom.

To evaluate the effectiveness of the proposed dual-level structural context collaborative framework, this paper selects a wide range of existing KGC methods as baselines, including TransE [1], DistMult [2], ComplEx [3], RotatE [9], KGTuner [35], UniGE [36], CompoundE [37], StAR [5], KG-S2S [38], SimKGC [6], GHN [39], HaSa [40], PEMLM [12], BMKGC [41], and ProgKGC [13]. Among them, TransE, SimKGC, ComplEx, and RotatE have been introduced in related work; the remaining methods are summarized as follows.

KGTuner [35] improves KGE training through efficient hyperparameter search. It first explores candidate hyperparameter configurations on small-scale subgraphs and then transfers promising configurations to the full KG for fine-tuning, thereby reducing search cost while obtaining better KGE configurations. UniGE [36] targets unified modeling of different geometric relation patterns in KGs. It represents diverse entity-relation structures with a unified geometric embedding framework, allowing the model to adapt to different relation patterns within the same representation space. CompoundE [37] models relations between entities by composing geometric transformations such as translation, rotation, and scaling. It represents complex relations as compositions of multiple basic geometric operations, thereby enhancing the expressiveness of embedding models for asymmetric and compositional relations.

StAR [5] is a structure-enhanced text representation learning method. It separately encodes query-side and entity-side text with a bi-encoder architecture and introduces triple structural information into textual representations, balancing inference efficiency and structural awareness. KG-S2S [38] formulates KGC as a sequence-to-sequence generation task and uses a pre-trained text generation model to generate the target

entity from the head entity and relation text. SimKGC [6] adopts a PLM-based bi-encoder framework and combines InfoNCE loss, in-batch negatives, pre-batch negatives, and self-negatives for efficient contrastive learning, making it a representative method in PLM-based KGC.

GHN [39] improves KGC training through generative hard negative mining, using a generative model to construct more confusing negative samples and strengthen discrimination among similar candidate entities. HaSa [40] starts from the trade-off between hard negatives and false negatives and proposes a hardness- and structure-aware contrastive learning strategy, which generates hard negatives while using graph structure to reduce training bias caused by potential false negatives. PEMLM [12] obtains semantic representations by pre-encoding entity textual descriptions and further fuses structural embeddings with pre-encoded semantic descriptions, reducing the training and inference cost of description-based models while improving prediction performance in low-resource scenarios.

BMKGC [41] builds a KGC model with a bilateral masked prompt mechanism. By jointly modeling head entity prediction and tail entity prediction directions, it enables the PLM to more fully exploit bidirectional semantic clues in triples. ProgKGC [13] proposes a progressive structure-enhanced semantic framework. It first builds PLM-based entity semantic representations and then gradually introduces a structural encoder and progressive training strategy to alleviate the coupling difficulty between semantic representation and graph structure modeling.

4.2 Entity Prediction Results

Table 4 reports the entity prediction results on WN18RR, FB15k-237, and Wikidata5M. The best results are shown in bold, the second-best results are underlined, and "–" indicates unreported results. Overall, DSC^{2}F achieves the best performance on all three datasets across MRR, Hits@1, Hits@3, and Hits@10, showing that the proposed framework provides consistent improvements over both embedding-based and PLM-based baselines. The gains on Hits@1 are particularly clear, indicating that DSC^{2}F is effective in ranking the correct entity at the top rather than only improving the general ranking order.

On WN18RR, DSC^{2}F obtains an MRR of 0.708 and improves over the strongest baseline ProgKGC by 0.020, 0.035, 0.014, and 0.002 on MRR, Hits@1, Hits@3, and Hits@10, respectively. The largest improvement appears on Hits@1, suggesting stronger top-ranked prediction ability on this relatively sparse dataset. On FB15k-237, DSC^{2}F also achieves the best results on all metrics, with an MRR of 0.365 and a Hits@10 of 0.551. Although the improvements on this dataset are more moderate, the consistent gains across all metrics indicate stable performance in a more complex multi-relation scenario.

The advantage becomes more pronounced on the large-scale Wikidata5M dataset. DSC^{2}F reaches 0.420 MRR and outperforms the second-best GHN by 0.056, 0.069, 0.055, and 0.046 on MRR, Hits@1, Hits@3, and Hits@10, respectively. This larger margin shows that DSC^{2}F scales well when the entity set is much larger and the candidate space is more challenging. Taken together, the results demonstrate that DSC^{2}F improves both strict ranking metrics and broader top-k retrieval metrics, with especially strong advantages in precise entity prediction.

To assess whether the improvements introduced by the proposed structural components over the SimKGC base model are statistically reliable, Table 5 reports two-tailed paired t-tests between DSC^{2}F and SimKGC. The paired observations are obtained from three runs using matched random seeds and identical dataset splits.

Table 4: Entity prediction results on WN18RR, FB15k-237, and Wikidata5M.

Category	Method	WN18RR				FB15k-237				Wikidata5M			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Embedding-based approach													
	TransE (2013)	0.243	0.043	0.441	0.532	0.279	0.198	0.376	0.441	0.253	0.170	0.311	0.392
	DistMult (2015)	0.444	0.412	0.470	0.504	0.281	0.199	0.301	0.446	0.253	0.209	0.278	0.334
	ComplEx (2016)	0.449	0.409	0.469	0.530	0.278	0.194	0.297	0.450	0.308	0.255	–	0.398
	RotatE (2019)	0.471	0.421	0.490	0.568	0.335	0.243	0.374	0.529	0.290	0.234	0.322	0.390
	KGTuner (2022)	0.481	0.438	0.499	0.556	0.345	0.252	0.381	0.534	–	–	–	–
	UniGE (2024)	0.491	0.447	0.512	0.563	0.343	0.257	0.375	0.523	–	–	–	–
	CompoundE (2023)	0.492	0.452	0.510	0.570	0.350	0.262	0.390	0.547	–	–	–	–
Text-based approach													
	StAR (2021)	0.398	0.238	0.487	0.698	0.288	0.195	0.313	0.480	–	–	–	–
	KG-S2S (2022)	0.572	0.529	0.595	0.663	0.337	0.255	0.374	0.496	–	–	–	–
	SimKGC (2022)	0.666	0.587	0.717	0.800	0.336	0.249	0.362	0.511	0.358	0.313	0.376	0.441
	GHN (2023)	0.678	0.596	0.719	0.821	0.339	0.251	0.364	0.518	0.364	0.317	0.380	0.453
	HaSa (2024)	0.535	0.446	0.587	0.711	0.301	0.218	0.325	0.482	–	–	–	–
	PEMLM (2024)	0.556	0.509	0.573	0.648	0.355	0.264	0.389	0.538	–	–	–	–
	BMKGC (2024)	0.669	0.590	0.720	0.807	0.332	0.247	0.365	0.514	–	–	–	–
	ProgKGC (2025)	0.688	0.597	0.739	0.834	0.344	0.255	0.375	0.523	–	–	–	–
	DSC^2F	**0.708**	**0.632**	**0.753**	**0.836**	**0.365**	**0.276**	**0.396**	**0.551**	**0.420**	**0.386**	**0.435**	**0.499**

Note: Bold values indicate the best results, and underlined values indicate the second-best results. H@1, H@3, and H@10 denote Hits@1, Hits@3, and Hits@10, respectively. The symbol "–" indicates that the corresponding result was not reported.

Table 5: Two-tailed paired t-test results for DSC^2F vs. SimKGC.

Dataset	Metric	t-Value	df	p-Value
WN18RR	MRR	43.2	2	<0.001
	Hits@1	87.1	2	<0.001
	Hits@3	89.6	2	<0.001
	Hits@10	90.5	2	<0.001
FB15k-237	MRR	34.0	2	<0.001
	Hits@1	59.1	2	<0.001
	Hits@3	98.8	2	<0.001
	Hits@10	101.2	2	<0.001

All comparisons yield $p < 0.001$, indicating that the improvements of DSC^2F over SimKGC are statistically significant on both datasets under the paired-run evaluation.

4.3 Ablation Study

To analyze the contribution of each component, this paper evaluates several variants of DSC^2F: w/o ISC removes both SNC and RAA, w/o SNC removes only neighborhood structural prompts, w/o RAA removes only relation-aware attention, and w/o BSC removes BRWR-driven batch construction while retaining instance-level structural modeling. Tables 6 and 7 show the results on WN18RR and FB15k-237, respectively.

Table 6: Ablation results of DSC^{2}F on WN18RR.

Method	SNC	RAA	BSC	MRR	H@1	H@3	H@10
w/o ISC	×	×	✓	0.674	0.607	0.723	0.822
w/o RAA	✓	×	✓	0.697	0.623	0.741	0.829
w/o SNC	×	✓	✓	0.676	0.608	0.727	0.824
w/o BSC	✓	✓	×	0.685	0.614	0.730	0.827
DSC^{2}F	✓	✓	✓	**0.708**	**0.632**	**0.753**	**0.836**

Note: The symbol ✓ indicates that the module is included, and × indicates that the module is removed. Bold values indicate the best results. H@1, H@3, and H@10 denote Hits@1, Hits@3, and Hits@10, respectively.

Table 7: Ablation results of DSC^{2}F on FB15k-237.

Method	SNC	RAA	BSC	MRR	H@1	H@3	H@10
w/o ISC	×	×	✓	0.354	0.266	0.387	0.539
w/o RAA	✓	×	✓	0.361	0.272	0.393	0.547
w/o SNC	×	✓	✓	0.355	0.267	0.387	0.540
w/o BSC	✓	✓	×	0.347	0.257	0.374	0.526
DSC^{2}F	✓	✓	✓	**0.365**	**0.276**	**0.396**	**0.551**

Note: The symbol ✓ indicates that the module is included, and × indicates that the module is removed. Bold values indicate the best results. H@1, H@3, and H@10 denote Hits@1, Hits@3, and Hits@10, respectively.

Table 6 shows that, on WN18RR, the full DSC^{2}F model achieves the best performance on all metrics, and removing any structural component leads to degradation. This indicates that SNC, RAA, and BSC provide complementary structural information rather than redundant effects. In particular, removing both SNC and RAA causes a clear drop, showing that batch-level structural constraints alone are insufficient to capture local topological evidence within each query.

Among the two instance-level submodules, removing SNC causes a larger decline than removing RAA, suggesting that neighborhood structural prompts are the main source of local structural information on WN18RR. RAA still contributes stable gains by modulating existing structural prompts according to the current relation, thereby improving structural-semantic alignment. Removing BSC also reduces all metrics, demonstrating that BRWR-driven batches provide additional structural discrimination pressure during training. Overall, the WN18RR results verify the complementarity of SNC, RAA, and BSC.

Table 7 reports the ablation results on FB15k-237. The full model again outperforms all variants on every metric, confirming the effectiveness of dual-level structural context in a more complex multi-relation KG. Removing BSC causes the most obvious degradation, indicating that BRWR-based structurally related batches are especially important for increasing negative-sample difficulty and learning structure-sensitive representations in dense multi-relation graphs.

Removing ISC also decreases performance, which shows that explicit instance-level modeling remains necessary even when batch-level structural constraints are retained. Consistent with WN18RR, removing SNC has a stronger effect than removing RAA, indicating that neighborhood prompts are the main source of instance-level structural information. However, the relative advantage of SNC is less pronounced on FB15k-237, likely because its more complex neighborhoods introduce additional noise. RAA provides stable gains by relation-conditionally filtering neighborhood information.

Overall, the ablation results show that DSC^{2}F benefits from the collaboration of SNC, RAA, and BSC. SNC supplies local neighborhood evidence, RAA improves relation-aware structural alignment, and BSC strengthens negative learning through topology-near batches. The results also suggest that SNC is more prominent on WN18RR, whereas BSC is more critical on FB15k-237, verifying that the proposed framework enhances structural awareness at both the encoding and training levels.

4.4 Structural Configuration Analysis

This section studies three key settings closely related to dual-level structural context: batch size, sampling strategy, and neighborhood size. These settings affect the number of in-batch negatives, the topological relevance of each batch, and the amount of local structural evidence available to the encoder.

4.4.1 Effect of Batch Size

Table 8 shows the performance of DSC^{2}F under different batch sizes. Overall, the optimal batch size is closely related to dataset scale and structural complexity. On WN18RR, the model achieves the best performance with a batch size of 1024; further increasing the batch size decreases performance, indicating that a moderate batch already provides sufficient structural negatives for a relatively sparse dataset with fewer relation types. Excessively large batches may introduce samples weakly related to the local subgraph and weaken batch-level structural consistency. On FB15k-237, the model reaches the best performance when the batch size increases to 3072, showing that more complex entity interactions and relation patterns require larger batches to provide sufficient structural hard negatives. This result indicates that a proper batch size balances negative-sample richness, structural relevance, and training noise, thereby improving the quality of batch-level structural context.

Table 8: Performance of DSC^{2}F under different batch sizes.

Dataset	Batch Size	MRR	H@1	H@3	H@10
WN18RR	512	0.703	0.616	0.747	0.831
	1024	**0.708**	**0.632**	**0.753**	**0.836**
	1536	0.704	0.628	0.751	0.831
	2048	0.695	0.616	0.742	0.827
FB15k-237	1024	0.332	0.247	0.360	0.520
	2048	0.347	0.252	0.368	0.526
	3072	**0.365**	**0.276**	**0.396**	**0.551**
	4096	0.357	0.267	0.384	0.543

Note: Bold values indicate the best results for each dataset. H@1, H@3, and H@10 denote Hits@1, Hits@3, and Hits@10, respectively.

4.4.2 Effect of Sampling Strategy

Table 9 shows the influence of different subgraph sampling strategies on DSC^{2}F. RWR returns to the starting node with a certain probability at each step and otherwise uniformly selects the next node from the current node's neighbors, which preserves local topological proximity but cannot distinguish the sampling value of different neighbors. BRWR introduces a neighbor-selection bias on top of RWR and adopts an inverse-degree transition distribution. BRWR_P uses a transition distribution proportional to node degree, while MCMC generates training subgraphs through Markov-chain state transitions.

Table 9: Performance of DSC^{2}F under different batch sampling strategies.

Dataset	Method	MRR	H@1	H@3	H@10
WN18RR	RWR	0.704	0.628	0.749	0.831
	BRWR	**0.708**	**0.632**	**0.753**	**0.836**
	BRWR_P	0.691	0.619	0.741	0.826
	MCMC	0.701	0.624	0.747	0.829
FB15k-237	RWR	0.359	0.271	0.391	0.543
	BRWR	**0.365**	**0.276**	**0.396**	**0.551**
	BRWR_P	0.350	0.265	0.384	0.536
	MCMC	0.342	0.258	0.377	0.528

Note: Bold values indicate the best results for each dataset. H@1, H@3, and H@10 denote Hits@1, Hits@3, and Hits@10, respectively.

From the perspective of batch-level structural context, BRWR performs best because it satisfies two requirements simultaneously. On the one hand, the restart mechanism ensures that the sampled subgraph remains centered on the current triple, so in-batch negatives stay topologically close to positives. On the other hand, inverse-degree bias reduces the dominance of high-degree entities and allows batches to cover more local structures related to long-tail or low-degree entities. Such a batch distribution is more likely than random or uniform walks to produce structurally similar but semantically different candidates, thereby imposing stronger structural discrimination pressure on the model.

4.4.3 Effect of Neighborhood Size

Fig. 2 shows the influence of the number of sampled neighbors on DSC^{2}F. When the neighborhood size is 0, the model does not use instance-level structural context and retains only batch-level structural context; in this case, the MRR values on WN18RR and FB15k-237 are 0.674 and 0.354, respectively. After neighborhood structure is introduced, both datasets improve, indicating that neighborhood prompts provide effective local topological evidence for instance-level structural context. On WN18RR, MRR reaches 0.708 when the neighborhood size is 16 and then becomes stable. This is mainly because WN18RR is sparse, and most entities have no additional useful neighbors when the sampling upper bound continues to increase. In contrast, FB15k-237 is denser and contains more complex relation interactions; therefore, a larger neighborhood size can cover more effective structural information and reaches the best MRR of 0.365 at size 64. Overall, the optimal neighborhood size is related to graph connectivity density: sparse graphs require only small neighborhoods, whereas dense graphs usually need larger neighborhood coverage to capture structural diversity.

The above structural configuration analyses show that DSC^{2}F is jointly affected by batch size, sampling strategy, and neighborhood size. Reasonable configurations can simultaneously enhance batch-level hard negative construction and instance-level structural evidence modeling, while the optimal settings of different datasets depend on graph sparsity, relation complexity, and entity connectivity density. These results further verify the effectiveness of dual-level structural context collaboration.

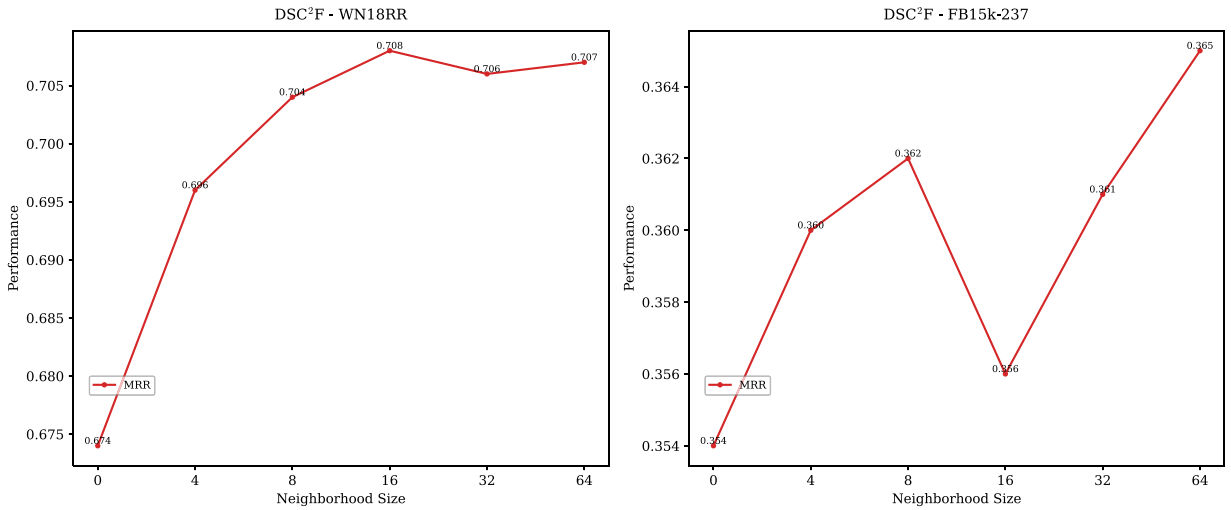

Figure 2: MRR of DSC^{2}F under different neighborhood sizes on WN18RR and FB15k-237.

5 Conclusion and Perspective

This paper proposes DSC^{2}F, a dual-level structural context collaborative framework for PLM-based KGC. At the instance level, SNC injects local neighborhood evidence into PLM encoding, and RAA aligns structural aggregation with relation semantics. At the batch level, BSC uses BRWR to construct topology-aware training batches and provides realistic hard negatives. Experiments on WN18RR, FB15k-237, and Wikidata5M demonstrate that the proposed framework consistently improves entity prediction performance. Ablation studies further show that SNC, RAA, and BSC are complementary and jointly enhance structural awareness.

Future work can extend this method in three directions. First, the current neighborhood prompt introduces a certain number of one-hop neighbors. Although this improves local structural awareness, it also brings additional computation and input-length overhead. More efficient neighbor injection strategies, such as adaptive neighbor selection and structural information compression, can be explored to reduce complexity while preserving structural expressiveness. Second, although DSC^{2}F effectively integrates structural information for KGC, its current relation-aware design relies on ID-based relation embeddings, which limits its ability to generalize to completely unseen relations and therefore weakens the zero-shot advantage provided by textual PLM representations. Developing a text-guided relation encoder that combines relation descriptions with structural signals is an important direction for supporting zero-shot relation generalization. Third, topology-aware batch construction can increase the risk of latent false negatives because real-world KGs are inherently incomplete. Although the current known-positive masking strategy excludes observed valid tails from the in-batch negative set, it cannot identify unobserved but valid facts. Future work will investigate uncertainty-aware negative reweighting and latent false-negative filtering to improve the robustness of structure-aware contrastive learning.

Acknowledgement: Not applicable.

Funding Statement: This work was supported in part by the Funds for Central-Guided Local Science & Technology Development (Grant No. 202407AC110005) Key Technologies for the Construction of a Whole-Process Intelligent Service System for Neuroendocrine Neoplasm; in part by the Funds for the Xingdian Talent Project of Yunnan Province, Key Technology Research and Application of Cross-Domain Automatic Business Collaboration in Smart Tourism (No. XYYC-CYCX-2022-0005); and in part by the Yunnan Provincial Department of Education's Enterprise-Proposed

Problem-Solving Project, "Research and Application Demonstration of Urban Low-Altitude IoT Intelligent Service System" (Project No. FWCY-QYCT2025001).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Jing Wang and Hao Li; methodology: Jing Wang and Hao Li; software: Jing Wang; validation and formal analysis: Tian Xia; data curation: Tian Xia; draft manuscript preparation: Jing Wang; supervision: Hao Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst. 2013;26:2787–95.
2. Yang B, SWt Y, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015; 2015 May 7–9; San Diego, CA, USA.
3. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. In: Proceedings of the International Conference on Machine Learning; 2016 Jun 19–24; New York, NY, USA. p. 2071–80.
4. Yao L, Mao C, Luo Y. KG-BERT: BERT for knowledge graph completion. arXiv:1909.03193. 2019.
5. Wang B, Shen T, Long G, Zhou T, Wang Y, Chang Y. Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of the Web Conference 2021; 2021 Apr 19–23; Ljubljana, Slovenia. p. 1737–48.
6. Wang L, Zhao W, Wei Z, Liu J. 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); 2022 May 22–27; Dublin, Ireland. p. 4281–94.
7. Li Q, Zhong Y, Qin Y. MoCoKGC: momentum contrast entity encoding for knowledge graph completion. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 14940–52.
8. Ko Y, Yang H, Kim T, Kim H. Subgraph-aware training of language models for knowledge graph completion using structure-aware contrastive learning. In: Proceedings of the ACM on Web Conference 2025; 2025 Apr 28–May 2; Sydney, Australia. p. 72–85.
9. Sun Z, Deng ZH, Nie JY, Tang J. RotatE: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations; 2019 May 6–9; New Orleans, LA, USA.
10. Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: Proceedings of the European Semantic Web: 15th International Conference, ESWC 2018; 2018 Jun 3–7; Heraklion, Greece. Cham, Switzerland: Springer; 2018. p. 593–607.
11. Vashishth S, Sanyal S, Nitin V, Talukdar P. Composition-based multi-relational graph convolutional networks. In: Proceedings of the International Conference on Learning Representations; 2020 Apr 26–30; Addis Ababa, Ethiopia.
12. Qiu C, Qian P, Wang C, Yao J, Liu L, Wei F, et al. Joint pre-encoding representation and structure embedding for efficient and low-resource knowledge graph completion. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 15257–69.
13. Li Z, Wu Y, Yuan Y, Wang J. ProgKGC: progressive structure-enhanced semantic framework for knowledge graph completion. In: Proceedings of the International Semantic Web Conference; 2025 Nov 2–6; Nara, Japan. Cham, Switzerland: Springer; 2025. p. 81–98.

14. Yao L, Peng J, Mao C, Luo Y. Exploring large language models for knowledge graph completion. In: Proceedings of the ICASSP 2025—2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2025 Apr 6–11; Hyderabad, India. p. 1–5.
15. Wei Y, Huang Q, Kwok JT, Zhang Y. KICGPT: large language model with knowledge in context for knowledge graph completion. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023; 2023 Dec 6–10; Singapore. p. 8667–83.
16. Yuan D, Zhou S, Chen X, Wang D, Liang K, Liu X, et al. Knowledge graph completion with relation-aware anchor enhancement. Proc AAAI Conf Artif Intell. 2025;39(14):15239–47. doi:10.1609/aaai.v39i14.33672.
17. Yang M, Yang C, Zhu J, Li J, Zhang J, Li Y, et al. SLiNT: structure-aware language model with injection and contrastive training for knowledge graph completion. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025; 2025 Nov 4–9; Suzhou, China. p. 13658–71.
18. Wang H, Song D, Wu Z, Tian Y, Yang P. Path-enhanced pre-trained language model for knowledge graph completion. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025; 2025 Nov 4–9; Suzhou, China. p. 4528–40.
19. Liu Y, Cao Y, Lin X, Shang Y, Wang S, Pan S. 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; 2025 Nov 4–9; Suzhou, China. p. 20981–95.
20. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901. doi:10.65525/svup.9788199778009.2026.224-230.
21. Li XL, Liang P. Prefix-tuning: optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long papers); 2021 Aug 1–6; Online. p. 4582–97.
22. Liu X, Ji K, Fu Y, Tam W, Du Z, Yang Z, et al. P-tuning: prompt tuning can be comparable to fine-tuning across scales and tasks. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short papers); 2022 May 22–27; Dublin, Ireland. p. 61–8.
23. Chen C, Wang Y, Sun A, Li B, Lam KY. Dipping PLMs sauce: bridging structure and text for effective knowledge graph completion via conditional soft prompting. In: Proceedings of the Findings of the association for computational linguistics: ACL 2023; 2023 Jul 9–14; Toronto, ON, Canada. p. 11489–503.
24. Jiang P, Agarwal S, Jin B, Wang X, Sun J, Han J. Text augmented open knowledge graph completion via pre-trained language models. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023; 2023 Jul 9–14; Toronto, ON, Canada. p. 11161–80.
25. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2014 Aug 24–27; New York, NY, USA. p. 701–10.
26. Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13–17; San Francisco, CA, USA. p. 855–64.
27. Ristoski P, Rosati J, Di Noia T, De Leone R, Paulheim H. Rdf2vec: RDF graph embeddings and their applications. Semant Web. 2019;10(4):721–52. doi:10.3233/sw-180317.
28. Tong H, Faloutsos C, Pan JY. Random walk with restart: fast solutions and applications. Knowl Inf Syst. 2008;14(3):327–46.
29. Cai L, Wang WY. KBGAN: adversarial learning for knowledge graph embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long papers); 2018 Jun 1–6; New Orleans, LA, USA. p. 1470–80.
30. Zhang Y, Yao Q, Chen L. Simple and automated negative sampling for knowledge graph embedding. VLDB J. 2021;30(2):259–85. doi:10.1007/s00778-020-00640-7.
31. Ahrabian K, Feizi A, Salehi Y, Hamilton WL, Bose AJ. Structure aware negative sampling in knowledge graphs. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; Online. p. 6093–101.

32. Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2018 Feb 2–7; New Orleans, LA, USA.
33. Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality; 2015 Jul 30; Beijing, China. p. 57–66.
34. Wang X, Gao T, Zhu Z, Zhang Z, Liu Z, Li J, et al. KEPLER: a unified model for knowledge embedding and pre-trained language representation. Trans Assoc Comput Linguist. 2021;9:176–94.
35. Zhang Y, Zhou Z, Yao Q, Li Y. Efficient hyper-parameter search for knowledge graph embedding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long papers); 2022 May 22–27; Dublin, Ireland. p. 2715–35.
36. Liu Y, Cao Z, Gao X, Zhang J, Yan R. Bridging the space gap: unifying geometry knowledge graph embedding with optimal transport. Proc ACM Web Conf. 2024;2024:2128–37.
37. Ge X, Wang YC, Wang B, Kuo CCJ. Compounding geometric operations for knowledge graph completion. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long papers); 2023 Jul 9–14; Toronto, ON, Canada. p. 6947–65.
38. Saxena A, Kochsiek A, Gemulla R. 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); 2022 May 22–27; Dublin, Ireland. p. 2814–28.
39. Qiao Z, Ye W, Yu D, Mo T, Li W, Zhang S. Improving knowledge graph completion with generative hard negative mining. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023; 2023 Jul 9–14; Toronto, ON, Canada. p. 5866–78.
40. Zhang H, Zhang J, Molybog I. HaSa: hardness and structure-aware contrastive knowledge graph embedding. Proc ACM Web Conf. 2024;2024:2116–27.
41. Kong Y, Fan C, Chen Y, Zhang S, Lv Z, Tao J. Bilateral masking with prompt for knowledge graph completion. In: Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024; 2024 Jun 16–21; Mexico City, Mexico. p. 240–9.