SLiNT: Structure-aware Language Model with Injection and Contrastive Training for Knowledge Graph Completion

Mengxue Yang[1], **Chun Yang**[1], **Jiaqi Zhu**[1,2,*], **Jiafan Li**[1,2]
Jingqi Zhang[1], **Yuyang Li**[1,3], **Ying Li**[1*]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China
{yangmengxue20@mails.ucas.ac.cn, zhujq@ios.ac.cn, liying21@ucas.ac.cn}

Abstract

Link prediction in knowledge graphs (KGs) requires integrating structural information and semantic context to infer missing entities. While large language models (LLMs) offer strong generative reasoning capabilities, their limited exploitation of structural signals often results in *structural sparsity* and *semantic ambiguity*, especially under incomplete or zero-shot settings. To address these challenges, we propose **SLiNT** (**S**tructure-aware **L**anguage model with **I**njection and co**N**trastive **T**raining), a modular framework that injects KG-derived structural context into a frozen LLM backbone with lightweight LoRA-based adaptation for robust link prediction. Specifically, **Structure-Guided Neighborhood Enhancement (SGNE)** retrieves pseudo-neighbors to enrich sparse entities and mitigate missing context; **Dynamic Hard Contrastive Learning (DHCL)** introduces fine-grained supervision by interpolating hard positives and negatives to resolve entity-level ambiguity; and **Gradient-Decoupled Dual Injection (GDDI)** performs token-level structure-aware intervention while preserving the core LLM parameters. Experiments on WN18RR and FB15k-237 show that SLiNT achieves superior or competitive performance compared with both embedding-based and generation-based baselines, demonstrating the effectiveness of structure-aware representation learning for scalable knowledge graph completion.

1 Introduction

Knowledge graphs (KGs) encode real-world facts as structured triples (h, r, t), where h and t denote the *head* and *tail* entities, respectively, and r is the relation connecting them. As a backbone for structured knowledge representation, KGs empower a variety of downstream applications such as question answering (Saxena et al., 2020), recommendation (Wang et al., 2019), and commonsense reasoning (Lin et al., 2019). However, real-world KGs are often incomplete, which motivates the task of *knowledge graph completion* (KGC), i.e., predicting missing entities or relations.

Traditional KGC methods such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), and RotatE (Sun et al., 2019) learn low-dimensional embeddings for entities and relations, and rank candidate triples based on geometric scoring functions. While these models perform well in dense regions of the KG, they often underperform on long-tail entities with sparse local neighborhoods. To mitigate this, some extensions incorporate textual features (Wang et al., 2021b) or graph-aware context (Vashishth et al., 2020), but still struggle with generalization and semantic discrimination.

Recent advances in large language models (LLMs) have introduced a new paradigm for knowledge graph completion (KGC), where pretrained models leverage semantic priors to generate missing entities from textualized queries (Lewis et al., 2020; Raffel et al., 2020; Xie et al., 2022; Saxena et al., 2022). To improve grounding, several strategies incorporate KG-derived signals into prompts. Instruction tuning (Liu et al., 2024) encodes relation semantics and output formats into natural language templates, while structural augmentation (Liu et al., 2025; Wei et al., 2024; Yang et al., 2025) use local subgraphs or structure-aware demonstrations to better align with graph context. Despite these strategies having shown promising improvements in generation controllability and KG-awareness, persistent limitations emerge, as illustrated in Figure 1, when examining the link prediction query $(?, \texttt{born\_in}, \texttt{Salzburg})$:

- **Challenge 1: Structural Sparsity** — KG-augmented LLMs rely on local subgraph context for grounding, yet many entities are poorly connected. In this case, sparse links around the gold entity "Wolfgang Amadeus

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13658–13671
November 4-9, 2025

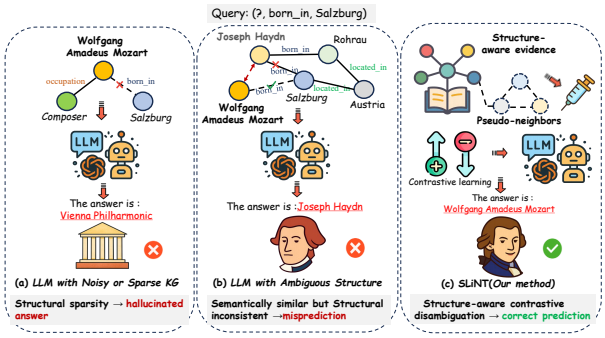

Figure 1: Motivating example for **SLiNT**. Given query $(?, \texttt{born\_in}, \texttt{Salzburg})$, (a) LLMs hallucinate due to sparse KG; (b) Semantic similarity overrides structural correctness, causing misprediction; (c) **SLiNT** disambiguates candidates via contrastive reasoning and structure injection.

Mozart" offer little structural support, thereby causing the model to hallucinate plausible but unsupported answers such as "Vienna Philharmonic." This reflects a critical failure: the generation collapses when the KG lacks sufficient structural cues.

- **Challenge 2: Semantic Ambiguity** — Even when structurally valid entities like "Wolfgang Amadeus Mozart" are retrieved, models may make wrong predictions by selecting semantically similar but incorrect alternatives such as "Joseph Haydn." This confusion arises because current LLMs favor surface-level similarity over structural alignment, lacking mechanisms to resolve fine-grained, relation-specific conflicts in entity semantics.

To tackle the aforementioned challenges, we propose **SLiNT** (**S**tructure-aware **L**anguage model with **I**njection and co**N**trastive **T**raining), a unified generative framework that explicitly integrates structural context and fine-grained supervision into a frozen LLM backbone with lightweight LoRA-based adaptation (Hu et al., 2022). To address Challenge 1, SLiNT introduces **Structure-Guided Neighborhood Enhancement (SGNE)**, which retrieves top-k_s pseudo-neighbors from pretrained KG embeddings and fuses them with attention to construct richer contextual representations for sparsely connected entities. To mitigate Challenge 2, we develop **Dynamic Hard Contrastive Learning (DHCL)**, which synthesizes interpolated hard positives and negatives based on semantic proximity and structural signals, encouraging the model to distinguish structurally coherent answers from semantically similar but misleading distractors. To bridge the gap between structural representations and language generation, we further design **Gradient-Decoupled Dual Injection (GDDI)**, which injects the enhanced structural representations into the frozen LLM backbone at the token level through prompt-based augmentation and substitution, while leveraging lightweight LoRA adaptation to maintain parameter efficiency. Together, these components enable SLiNT to perform robust link prediction under both sparse and ambiguous KG scenarios, while maintaining generation fluency, structural faithfulness, and parameter efficiency. Our main contributions are summarized as follows:

- We propose **SLiNT**, the first structure-aware generative framework that jointly integrates pseudo-neighbor enhancement, contrastive disambiguation, and token-level structure injection into a frozen LLM backbone for link prediction.
- We introduce two novel techniques to realize the framework: **DHCL**, for structure-aware contrastive learning, and **GDDI**, a lightweight gradient-decoupled injection mechanism.
- We empirically show that SLiNT achieves superior or competitive performance on two standard benchmarks, while maintaining robustness in sparse and ambiguous KG scenarios.

2 Related Work

Prior work on knowledge graph completion (KGC) can be broadly categorized into two paradigms: embedding-based models and generation-based models.

Embedding-based KGC. Classical models such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), and RotatE (Sun et al., 2019) encode entities and relations into continuous vector spaces, scoring triples based on distance or semantic compatibility. While efficient and interpretable, these models depend heavily on dense local structures and struggle with long-tail or sparsely connected entities. Later extensions incorporate auxiliary textual (Wang et al., 2021b) or structural (Vashishth et al., 2020) information to improve robustness, but remain limited in handling diverse or ambiguous semantics.

Generation-based KG Completion. Unlike embedding-based methods that learn entity and relation vectors, generation-based approaches formulate KG completion as a text generation task. Early works such as KGT5 (Saxena et al., 2022), GenKGC (Xie et al., 2022) and KG-S2S (Chen et al., 2022) recast triple prediction into sequence-to-sequence learning, enabling flexible reasoning over natural language. Later models, including GS-KGC (Yang et al., 2025) and FtG (Liu et al., 2025), incorporate structural features from subgraphs to provide auxiliary context, while KICGPT (Wei et al., 2024) and DIFT (Liu et al., 2024) adopt instruction tuning and in-context demonstrations to improve generation quality. These approaches significantly advance LLM-based KGC by leveraging pretrained language priors. However, they typically lack mechanisms to model structural ambiguity or decision boundaries, limiting their performances in sparse or confusing regions.

3 Methodology

SLiNT tackles two key challenges in knowledge graph completion (KGC)—*structural sparsity* and *semantic ambiguity*—via three modules: structure-guided neighborhood enhancement, contrastive learning, and structure-aware injection. As shown in Figure 2, the pipeline starts from pretrained KG embeddings, sequentially applies enhancement and contrastive supervision, then injects structure-derived signals into a frozen LLM backbone, while fine-tuning only the lightweight LoRA adapters for efficient adaptation.

Problem Formulation. We formalize the task as link prediction over a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denotes a set of factual triples. Given an incomplete query $q = (h, r, ?)$ or $(?, r, t)$, the objective is to find the missing entity, either the head h or the tail t, from the entity set $\mathcal{E}$. Following DIFT, we adopt a two-stage formulation. A pretrained KG embedding model M_E is first used to rank all candidate entities, producing a top-m list:

$$\mathcal{C}(q) = \text{Top-m}(M_E(q)) = [e_1, e_2, \ldots, e_m], \quad (1)$$

where each candidate e_i is associated with an embedding $\mathbf{e}_i \in \mathbb{R}^d$, and the query is represented by $\mathbf{q} \in \mathbb{R}^d$. These structural representations are used as inputs to the subsequent SLiNT modules.

3.1 Structure-Guided Neighborhood Enhancement (SGNE)

To address structural sparsity, SGNE enhances each input embedding, either a query $\mathbf{q} \in \mathbb{R}^d$ or a candidate entity $\mathbf{e}_i \in \mathbb{R}^d$, by aggregating its top-k_s *structural pseudo-neighbors*, i.e., nearest neighbors in the pretrained KG embedding space rather than true KG neighbors. We retrieve these pseudo-neighbors from a global entity pool $\mathcal{E} \in \mathbb{R}^{N \times d}$ based on cosine similarity:

$$\mathcal{N}_p(\mathbf{x}) = \text{Top-}k_s\left(\cos(\mathbf{x}, \mathcal{E})\right), \quad \mathbf{x} \in \{\mathbf{q}, \mathbf{e}_i\}. \quad (2)$$

Let $\mathbf{E}_n^{(x)} \in \mathbb{R}^{k_s \times d}$ denote the corresponding embedding matrix of pseudo-neighbors. We project both the input and its neighbors into a shared latent space using a learnable matrix $W_{\text{in}} \in \mathbb{R}^{d \times h}$, followed by a SiLU activation ϕ (Elfwing et al., 2018):

$$\mathbf{h}^{(x)} = \phi(W_{\text{in}}\mathbf{x}), \quad \mathbf{H}_n^{(x)} = \phi(W_{\text{in}}\mathbf{E}_n^{(x)}). \quad (3)$$

We concatenate the input and neighbor representations, apply the multi-head attention:

$$\mathbf{Z}^{(x)} = \text{MultiHeadAttn}\left(\mathbf{h}^{(x)} \,\|\, \mathbf{H}_n^{(x)}\right). \quad (4)$$

The enhanced representation is obtained by extracting the first token and projecting it back to the output dimension via $W_{\text{out}} \in \mathbb{R}^{h \times d'}$:

$$\tilde{\mathbf{x}} = W_{\text{out}}\mathbf{Z}_0^{(x)}, \quad \tilde{\mathbf{x}} \in \{\tilde{\mathbf{q}}, \tilde{\mathbf{e}}_i\}. \quad (5)$$

The final outputs $\tilde{\mathbf{q}}, \tilde{\mathbf{e}}_i \in \mathbb{R}^{d'}$ are used for contrastive learning and token-level generation in subsequent modules.

3.2 Dynamic Hard Contrastive Learning (DHCL)

While SGNE enhances structural representations, it lacks supervision to distinguish structurally coherent entities from semantically similar but structurally divergent distractors. To address this, we propose Dynamic Hard Contrastive Learning (DHCL), a structure-sensitive contrastive objective that promotes fine-grained discrimination in the structural space. Rather than comparing raw entity pairs, DHCL interpolates between the query and structure-derived prototypes to generate boundary-level hard positives and negatives, simulating ambiguous decisions near structural margins. This encourages the model to favor structurally valid answers and suppress misleading semantic lookalikes. The full procedure is shown in Algorithm 1.

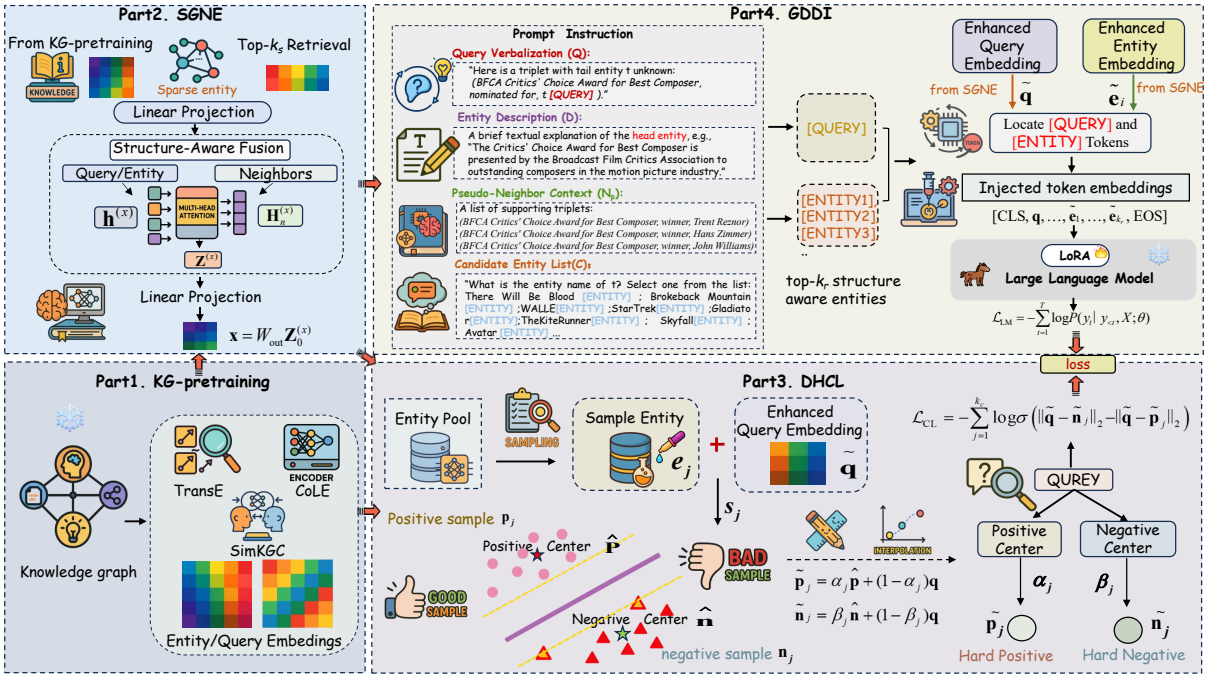

Figure 2: **Overview of the SLiNT framework.** It consists of **SGNE** for neighbor-enhanced fusion, **DHCL** for contrastive augmentation, and **GDDI** for structural injection with LoRA fine-tuning.

Algorithm 1: Dynamic Hard Contrastive Learning (DHCL)

Input: Enhanced query $\tilde{\mathbf{q}} \in \mathbb{R}^{d'}$;
Entity pool $\mathcal{E} \in \mathbb{R}^{N \times d'}$;
Sample size N_c;
Contrastive sample size k_c
Output: Contrastive loss $\mathcal{L}_{\text{CL}}$

Sample N_c entities $\{\mathbf{e}_j\}_{j=1}^{N_c} \subset \mathcal{E}$;
Compute cosine similarity $s_j \leftarrow \cos(\tilde{\mathbf{q}}, \mathbf{e}_j)$;
Select positives $\mathcal{P}_c \leftarrow \text{Top-k}_c^{\text{high}}(\{s_j\})$;
Select negatives $\mathcal{N}_c \leftarrow \text{Top-k}_c^{\text{low}}(\{s_j\})$;
Compute prototype centers $\hat{\mathbf{p}}, \hat{\mathbf{n}}$;
Interpolate hard samples $\tilde{\mathbf{p}}_j, \tilde{\mathbf{n}}_j$;
Compute contrastive loss $\mathcal{L}_{\text{CL}}$;
return $\mathcal{L}_{\text{CL}}$;

Given a query embedding $\mathbf{q}$, enhanced via SGNE as $\tilde{\mathbf{q}}$, we randomly sample N_c entities $\{\mathbf{e}_j\}$ from the global entity pool $\mathcal{E}$, and compute their cosine similarities:

$$s_j = \cos(\tilde{\mathbf{q}}, \mathbf{e}_j) = \frac{\tilde{\mathbf{q}}^\top \mathbf{e}_j}{\|\tilde{\mathbf{q}}\| \cdot \|\mathbf{e}_j\|}. \tag{6}$$

Hard Sample Mining. To provide contrastive supervision, the top-k_c most similar and least similar entities are selected:

$$\mathcal{P}_c = \text{Top-}k_c^{\text{high}}(\{s_j\}), \quad \mathcal{N}_c = \text{Top-}k_c^{\text{low}}(\{s_j\}), \tag{7}$$

where $\mathcal{P}_c = \{\mathbf{p}_j\}$ and $\mathcal{N}_c = \{\mathbf{n}_j\}$ denote the contrastive positive and negative entity sets, respectively. The hyperparameter k_c specifies the number of hard samples used for contrastive training.

Prototype Construction. To obtain representative prototypes for contrastive learning, we compute weighted centers:

$$\hat{\mathbf{p}} = \frac{\sum_j w_j^+ \mathbf{p}_j}{\sum_j w_j^+}, \quad \hat{\mathbf{n}} = \frac{\sum_j w_j^- \mathbf{n}_j}{\sum_j w_j^-}, \tag{8}$$

where $w_j^+, w_j^- \in \mathbb{R}^+$ are sampled from a uniform distribution.

Interpolation. To generate boundary-sensitive examples, interpolation is performed between the query and the prototype centers:

$$\begin{aligned} \tilde{\mathbf{p}}_j &= \alpha_j \hat{\mathbf{p}} + (1 - \alpha_j)\mathbf{q}, \\ \tilde{\mathbf{n}}_j &= \beta_j \hat{\mathbf{n}} + (1 - \beta_j)\mathbf{q}, \end{aligned} \tag{9}$$

where $\alpha_j, \beta_j \in [0, 1]$ are interpolation coefficients sampled uniformly. We constrain the coefficients to $\alpha_j, \beta_j \sim \mathcal{U}(0.3, 0.7)$ in practice. These synthetic samples approximate near-boundary contrasts in the structure space.

Contrastive Loss. We then optimize a margin-based contrastive loss over interpolated examples:

$$\mathcal{L}_{\text{CL}} = -\sum_{j=1}^{k_c} \log \sigma \left(\|\tilde{\mathbf{q}} - \tilde{\mathbf{n}}_j\|_2 - \|\tilde{\mathbf{q}} - \tilde{\mathbf{p}}_j\|_2 \right). \tag{10}$$

This encourages the model to align query representations with structurally consistent entities while pushing away structurally incompatible ones, improving fine-grained discrimination in structure-aware settings.

3.3 Gradient-Decoupled Dual Injection (GDDI)

We propose Gradient-Decoupled Dual Injection (GDDI), a dual mechanism that incorporates structure-enhanced representations into a frozen LLM backbone through *prompt-level augmentation* and *token-level injection*, while fine-tuning only the lightweight LoRA adapters. This design allows the model to leverage KG-derived context efficiently, without updating the full set of LLM parameters.

Prompt Construction. For each query triple $q = (h, r, ?)$ or $(?, r, t)$, we construct a generation prompt $\mathcal{P}(q)$ by concatenation:

$$\mathcal{P}(q) = [\mathcal{Q}; \mathcal{D}; \mathcal{N}_p; \mathcal{C}], \tag{11}$$

Where $\mathcal{Q}$ is a natural language verbalization of the query (e.g., "(BFCA Critics' Choice Award for Best Composer, nominated for, ?)"), $\mathcal{D}$ provides a brief textual description of the known entity (either head or tail), $\mathcal{N}_p$ contains pseudo-neighbor triplets retrieved by SGNE, which are constructed by mapping structural pseudo-neighbors (retrieved in the embedding space) back to KG triples, and $\mathcal{C}$ lists the top-m ranked candidates from the KG embedding model. An illustrative example from the FB15k-237 dataset is provided in Figure 2.

Token-Level Injection. After constructing the prompt, we identify the token positions of the `[QUERY]` and top-k_r `[ENTITY]` markers. We inject SGNE-enhanced embeddings at these locations:

$$\mathbf{E}_{\text{input}}[p_{\text{query}}] = \tilde{\mathbf{q}}, \quad \mathbf{E}_{\text{input}}[p^{(i)}_{\text{entity}}] = \tilde{\mathbf{e}}_i, \tag{12}$$

where $\tilde{\mathbf{q}}, \tilde{\mathbf{e}}_i \in \mathbb{R}^{d'}$ are structure-aware embeddings, and $p_{\text{query}}, p^{(i)}_{\text{entity}}$ denote the corresponding token indices.

Training. We adapt the frozen LLM backbone via parameter-efficient LoRA (Hu et al., 2022), where only LoRA adapters are trainable. This dual mechanism strengthens the model's capacity to integrate structure-aware representations during generation, enabling more accurate entity disambiguation in sparse and ambiguous contexts.

3.4 Training Objective

The training objective combines language modeling with structure-aware contrastive supervision:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{CL}}, \tag{13}$$

where $\lambda \in \mathbb{R}^+$ balances the generation loss $\mathcal{L}_{\text{LM}}$ and the contrastive loss $\mathcal{L}_{\text{CL}}$. The language modeling loss is defined as:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, X; \theta), \tag{14}$$

where y_t is the target token at timestep t, $y_{<t}$ is the partial output sequence, X is the structure-injected input, and θ denotes the LLM parameters. This ensures that the generation is conditioned not only on textual prompts but also on injected structural signals. Detailed theoretical analyses of retrieval complexity, the stability of the contrastive loss, and structural injection alignment are provided in Appendix A.1, A.3, and A.4.

4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of **SLiNT** on two widely used knowledge graph completion (KGC) benchmarks: **FB15k-237** (Toutanova et al., 2015) and **WN18RR** (Dettmers et al., 2018). The statistics analysis of these datasets is provided in Appendix B. Our experiments aim to answer the following research questions:

- **RQ1**: Does SLiNT outperform state-of-the-art embedding-based and generation-based KGC methods?
- **RQ2**: What are the individual contributions of SGNE, DHCL, and GDDI to the overall performance?
- **RQ3**: How robust is SLiNT under low-resource or structurally incomplete KG scenarios?

4.1 Experimental Setup

Baselines. We compare **SLiNT** with two categories of methods: (1) embedding-based models such as TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), and others; and (2) generation-based models including DIFT (Liu et al., 2024), KICGPT (Wei et al., 2024), and others. A full list of baselines is provided in Table 1.

Backbone Configurations. We instantiate SLiNT with three representative structural backbones: (1) **SLiNT + TransE** uses a translational embedding model (Bordes et al., 2013) that captures relational regularities through distance-based scoring, representing a lightweight and widely-used baseline for structure-only supervision; (2) **SLiNT + SimKGC** builds on contrastive representation learning (Wang et al., 2022), enhancing entity discrimination by aligning positive pairs and pushing apart hard negatives; (3) **SLiNT + CoLE** leverages a hybrid encoder (Liu et al., 2022) combining local neighborhood information with pre-trained textual features, providing rich structure-semantic fusion.

Evaluation Metrics. We adopt standard evaluation metrics commonly used in knowledge graph completion tasks, including **Mean Reciprocal Rank (MRR)** and **Hits@K** (with $K = 1, 3, 10$). **MRR** measures the average inverse rank of the correct entity across all test queries, providing a fine-grained assessment of ranking performance. **Hits@K** reports the proportion of test queries for which the correct entity appears within the top-K ranked candidates, reflecting the model's ability to retrieve relevant entities. Higher MRR and Hits@K scores indicate better predictive accuracy. These metrics are computed under the standard filtered setting, where corrupted triples that already exist in the KG are excluded during ranking to ensure fair evaluation.

4.2 Implementation Details

We implement SLiNT using PyTorch with mixed-precision training on 8×64GB MetaX GPUs (performance comparable to A100s). All experiments leverage frozen LLaMA-7B [1] with pre-trained KG embeddings (TransE, SimKGC, CoLE), and employ LoRA for lightweight adaptation. Key training hyperparameters include a batch size of 64, a learning rate of 2×10^{-5}, and contrastive loss weight $\lambda = 0.5$. Further configuration details are provided in Appendix C.

[1] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

4.3 Main Results (RQ1)

We evaluate SLiNT on FB15k-237 and WN18RR, comparing it against two major categories of methods: *embedding-based* models and *generation-based* models. Results are reported in Table 1.

Overall Performance. SLiNT achieves superior or competitive performance across both datasets. On **FB15k-237**, **SLiNT + CoLE** attains the highest MRR (0.443) and strong Hits@1 (0.368), while slightly trailing NBFNet in Hits@10 (0.599 vs. 0.591). On **WN18RR**, **SLiNT + SimKGC** yields the best MRR (0.691) and Hits@1 (0.626), with Hits@10 of 0.805, outperforming other generation-based baselines. Although NCRL achieves a higher Hits@10 (0.850), SLiNT + SimKGC maintains the best overall balance with superior MRR. Compared with vanilla LLaMA variants, SLiNT delivers notable MRR gains: +0.205 on FB15k-237 and +0.252 on WN18RR, highlighting the benefits of structure-aware injection and contrastive learning.

Comparison with Prior Methods. Embedding-based methods such as NBFNet and SimKGC perform well on WN18RR but underperform on long-tail relations on FB15k-237. Generation-based baselines like DIFT and KICGPT improve semantic controllability, but their reliance on template-based augmentation limits their robustness. SLiNT outperforms all prior generative models under identical KG embeddings (e.g., CoLE), highlighting its superior capacity to capture structure-aware semantics.

SLiNT Variants. SLiNT yields consistent improvements across all KG embeddings, TransE, SimKGC, and CoLE, validating its robustness and *plug-and-play* compatibility with frozen LLM backbone. Rather than relying on any specific encoder, SLiNT adapts flexibly to different structural priors. Case studies in Appendix D further illustrate how it resolves fine-grained ambiguities through structure-aware supervision.

4.4 Ablation Study (RQ2)

To assess the contribution of each component in SLiNT, we conduct ablation experiments on FB15k-237 and WN18RR using CoLE embeddings. Table 2 reports the results when removing

Models	FB15K-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Embedding-based								
TransE (Bordes et al., 2013)	0.312	0.212	0.354	0.510	0.225	0.016	0.403	0.521
RotatE (Sun et al., 2019)	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
TuckER (Balazevic et al., 2019)	0.358	0.266	0.394	0.544	0.470	0.443	0.482	0.526
Neural-LP (Yang et al., 2017)	0.237	0.173	0.259	0.361	0.381	0.368	0.386	0.408
NCRL (Cheng et al., 2023)	0.300	—	—	0.473	0.670	0.563	—	**0.850**
CompGCN (Vashishth et al., 2020)	0.355	0.264	0.390	0.535	0.479	0.443	0.494	0.546
HittER (Chen et al., 2021)	0.373	0.279	0.409	0.558	0.503	0.462	0.516	0.584
NBFNet (Zhu et al., 2021)	0.415	0.321	0.454	**0.599**	0.551	0.497	0.573	0.666
KG-BERT (Yao et al., 2019)	—	—	—	0.420	0.216	0.041	0.302	0.524
StAR (Wang et al., 2021a)	0.365	0.266	0.404	0.562	0.551	0.459	0.594	0.732
MEM-KGC (Choi et al., 2021)	0.346	0.253	0.381	0.531	0.557	0.475	0.604	0.704
SimKGC (Wang et al., 2022)	0.338	0.252	0.364	0.511	0.671	0.595	0.719	0.802
CoLE (Liu et al., 2022)	0.389	0.294	0.429	0.572	0.593	0.538	0.616	0.701
Generation-based								
GenKGC(Xie et al., 2022)	—	0.192	0.355	0.439	—	0.287	0.403	0.535
KGT5 (Saxena et al., 2022)	0.276	0.210	—	0.414	0.508	0.487	—	0.544
KG-S2S (Chen et al., 2022)	0.336	0.257	0.373	0.498	0.574	0.531	0.595	0.661
$\text{ChatGPT}_{\text{one-shot}}$ (OpenAI, 2023)	—	0.267	—	—	—	0.212	—	—
KICGPT (Wei et al., 2024)	0.412	0.327	0.448	0.581	0.564	0.478	0.612	0.677
LLaMA + TransE (Liu et al., 2024)	0.232	0.080	0.321	0.502	0.202	0.037	0.360	0.516
LLaMA + SimKGC (Liu et al., 2024)	0.236	0.074	0.335	0.503	0.391	0.065	0.695	0.798
LLaMA + CoLE (Liu et al., 2024)	0.238	0.087	0.387	0.561	0.374	0.117	0.602	0.697
DIFT + TransE (Liu et al., 2024)	0.389	0.322	0.408	0.525	0.491	0.462	0.496	0.560
DIFT + SimKGC (Liu et al., 2024)	0.402	0.338	0.418	0.528	0.686	0.616	0.730	0.806
DIFT + CoLE (Liu et al., 2024)	0.439	0.364	0.468	0.586	0.617	0.569	0.638	0.708
SLiNT + TransE	0.395	0.329	0.416	0.522	0.506	0.482	0.508	0.567
SLiNT + SimKGC	0.416	0.355	0.433	0.529	**0.691**	**0.626**	**0.731**	0.805
SLiNT + CoLE	**0.443**	**0.368**	**0.472**	0.591	0.626	0.578	0.646	0.718

Table 1: Link prediction results on FB15k-237 and WN18RR. Best results are in **bold** and second-best are underlined. We reproduce the results of TransE, SimKGC, and CoLE using their source codes and hyperparameters. The results of other baselines are obtained from their respective original papers.

SGNE, DHCL, or GDDI.

On FB15k-237, removing SGNE leads to a noticeable drop in MRR (0.443 → 0.429), highlighting the value of pseudo-neighbor fusion for structural enrichment. The absence of DHCL causes the largest decline in Hits@1 (0.368 → 0.329), underscoring its role in differentiating close structural candidates and reinforcing fine-grained decision boundaries through structure-aware contrastive training. Removing GDDI also degrades performance, albeit moderately, indicating that token-level structure injection provides complementary gains. Similar results appear on WN18RR, where disabling DHCL again leads to the greatest drop in Hits@1 (0.578 → 0.546), confirming its central role in optimizing entity-level decision boundaries. The consistent declines when omitting SGNE or GDDI further support the necessity of all the three components. Overall, these results demonstrate that each module contributes uniquely to SLiNT's effectiveness, and their integration is essential for accurate and structure-aware link prediction in challenging KG scenarios.

Config	FB15k-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Full	**0.443**	**0.368**	**0.472**	**0.591**	**0.626**	**0.578**	**0.646**	**0.718**
w/o SGNE	0.429	0.342	0.453	0.572	0.612	0.560	0.630	0.707
w/o DHCL	0.419	0.329	0.444	0.564	0.606	0.546	0.621	0.705
w/o GDDI	0.433	0.352	0.457	0.577	0.615	0.567	0.638	0.708

Table 2: Ablation results of **SLiNT** on FB15k-237 and WN18RR using CoLE. Each module contributes to overall performance.

4.5 Robustness Analysis (RQ3)

We evaluate SLiNT's robustness under two common types of knowledge graph sparsity: limited training supervision and incomplete struc-

tural connectivity. All experiments are conducted with **SLiNT+CoLE**. Specifically, we simulate low-resource settings by (1) reducing the training data to 80%, and (2) randomly removing 10% of KG edges. As shown in Figure 3, SLiNT maintains strong MRR under both conditions across FB15k-237 and WN18RR. Performance drops are marginal, confirming its stability against supervision loss and structural noise.

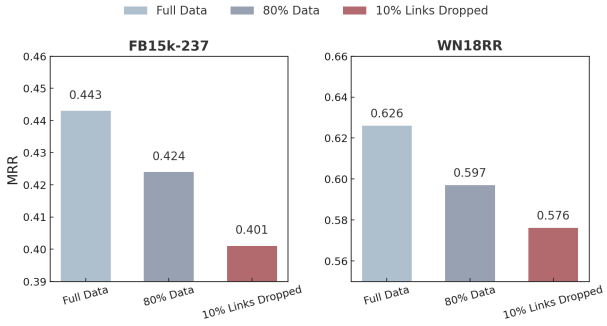

Figure 3: Performance of **SLiNT + CoLE** under limited supervision and structural incompleteness.

We further assess robustness under degree sparsity. Notice that both datasets exhibit long-tail degree distributions. As shown in Table 3, 36.7% of entities in WN18RR and 22.3% in FB15k-237 fall within the bottom 20% of degree, motivating our structure-guided design in SLiNT.

Dataset	#Ent.	AvgDeg	LowDeg (%)	Max/Min
WN18RR	40,943	4.2	36.7	221 / 1
FB15k-237	14,541	27.8	22.3	4,622 / 1

Table 3: Structural sparsity statistics. "LowDeg (%)" denotes the proportion of entities in the bottom 20% of degree.

We then evaluate SLiNT on test queries whose gold entities belong to this low-degree group. As shown in Table 4, SLiNT consistently outperforms strong baselines across both datasets, highlighting its superior generalization to structurally under-connected entities. These findings support the effectiveness of SGNE and DHCL in real-world incomplete graphs.

4.6 Further Analysis

Effect of Encoder Quality. SLiNT supports diverse structural encoders, and its performance can be further enhanced with more expressive ones. As shown in Table 1 and Figure 4, models using more powerful encoders such as CoLE consistently

Model	FB15k-237			WN18RR		
	H@1	H@10	MRR	H@1	H@10	MRR
LLaMA + CoLE	0.032	0.438	0.159	0.041	0.598	0.257
DIFT + CoLE	0.281	0.497	0.379	0.502	0.646	0.571
SLiNT + CoLE	**0.297**	**0.511**	**0.391**	**0.517**	**0.658**	**0.573**

Table 4: Performance on low-degree entities (bottom 20% degree group) from FB15k-237 and WN18RR.

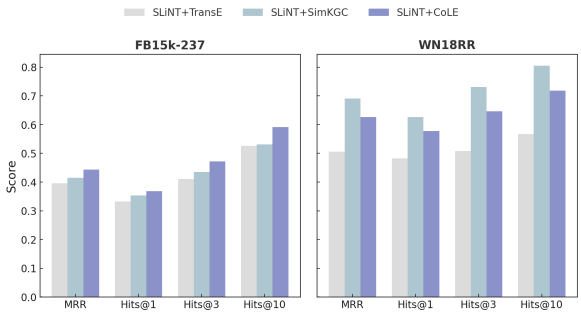

Figure 4: Comparing the performance of **SLiNT** with different structural encoders on FB15k-237 and WN18RR.

achieve higher performance. Notably, **SLiNT + CoLE** achieves the best MRR on FB15k-237, while **SLiNT + SimKGC** performs best on WN18RR, outperforming all baseline methods on their respective benchmarks. Even when paired with a simpler encoder like TransE, SLiNT surpasses several strong models, including LLaMA + TransE and SimKGC. These results demonstrate SLiNT's robustness to encoder quality and its ability to extract meaningful knowledge even from shallow embeddings.

Effect of Top-k_s Neighbors. We evaluate how the number of structural neighbors ($k_s \in \{1, 3, 5, 10\}$) in SGNE affects performance across different encoders. As shown in Figure 5, more expressive encoders like CoLE benefit steadily from increasing k_s, peaking at $k_s = 5$. SimKGC shows similar trends but remains more stable. For weaker encoders like TransE, performance improves up to k_s=5 but drops at k_s=10 due to noisy neighbors. These findings suggest a trade-off: too few neighbors fail to capture the structure, while too many introduce noise, especially under less robust encoders. SGNE remains effective across these settings, with k_s=5 serving as a balanced default.

Effect of Contrastive Loss Weight. To assess the impact of contrastive supervision, we evaluate SLiNT under varying contrastive weights $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$ (Table 5). Results show that the optimal λ is dataset-specific: on FB15k-237, the best MRR and Hits@k scores are achieved

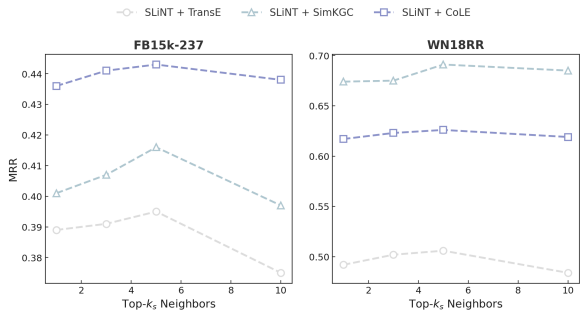

Figure 5: MRR comparison under varying Top-k_s neighbor sizes for **SLiNT** with different structural encoders on FB15k-237 and WN18RR.

λ	FB15k-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
0.1	0.438	0.354	0.462	0.580	0.617	0.571	0.639	0.709
0.3	**0.443**	**0.365**	**0.467**	**0.589**	0.619	0.574	0.637	0.708
0.5	0.437	0.352	0.459	0.582	**0.626**	**0.578**	**0.646**	**0.718**
0.7	0.421	0.336	0.446	0.570	0.619	0.571	0.640	0.709

Table 5: Performance of **SLiNT + CoLE** under varying contrastive loss weights λ on FB15k-237 and WN18RR.

at $\lambda = 0.3$, while WN18RR reaches its peak at $\lambda = 0.5$. This discrepancy likely reflects structural differences: WN18RR is more relation-regular with clearer decision boundaries, making it more receptive to contrastive supervision; In contrast, FB15k-237 contains more semantically overlapping relations, where overly strong contrastive signals may hinder generalization. Overall, these findings underscore the importance of balancing contrastive and generative objectives. Underweighting reduces the benefits of contrastive learning, while overweighting may destabilize training.

5 Conclusion

We present **SLiNT**, a structure-aware generative framework for knowledge graph completion that injects structure-derived evidence from KG embeddings into a frozen LLM backbone with lightweight LoRA-based adaptation. SLiNT integrates three complementary modules: SGNE for neighborhood enhancement, DHCL for dynamic contrastive supervision, and GDDI for token-level structure injection. Experiments on FB15k-237 and WN18RR demonstrate that SLiNT achieves superior or competitive performance, outperforming both embedding-based and generation-based baselines across multiple metrics, while maintaining robustness under structural sparsity and achieving efficient adaptation without full-model fine-tuning.

Limitations

While **SLiNT** achieves strong performance on standard KGC benchmarks, it mainly leverages structure-derived signals from pretrained KG embeddings. This limits its applicability to scenarios requiring multimodal cues (e.g., images and temporal dynamics). Future work could extend SGNE to incorporate multimodal retrieval and design adaptive injection strategies for better generalization and efficiency in diverse real-world settings.

Ethical Consideration

We use only publicly available datasets, which contain no private or sensitive information. No human subjects or annotation workers were involved. Our model incorporates open-source large language models (LLMs) for reasoning. While we do not fine-tune on sensitive content, we acknowledge potential risks of misuse or generation bias in downstream applications.

Acknowledgments

We would like to thank the reviewers and area chairs for their valuable feedback and constructive suggestions, which helped improve this work. This research was supported in part by the MIIT Project on Industrial Real-Time Database Based on Next-Generation Information Technology (TC210804D) and the CAS Project for Young Scientists in Basic Research (YSBR-040).

References

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 4005–4017. International Committee on Computational Linguistics.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. Hitter: Hierarchical transformers for knowledge graph embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10395–10407. Association for Computational Linguistics.

Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Bonggeun Choi, Daesik Jang, and Youngjoong Ko. 2021. MEM-KGC: masked entity model for knowledge graph completion with pre-trained language model. *IEEE Access*, 9:132025–132032.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839. Association for Computational Linguistics.

Ben Liu, Jihai Zhang, Fangquan Lin, Cheng Yang, and Min Peng. 2025. Filter-then-generate: Large language models with structure-text adapter for knowledge graph completion. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 11181–11195. Association for Computational Linguistics.

Yang Liu, Zequn Sun, Guangyao Li, and Wei Hu. 2022. I know what you do not know: Knowledge graph embedding via co-distillation learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1329–1338. ACM.

Yang Liu, Xiaobin Tian, Zequn Sun, and Wei Hu. 2024. Finetuning generative large language models with discrimination instructions for knowledge graph completion. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part I*, volume 15231 of *Lecture Notes in Computer Science*, pages 199–217. Springer.

David A. McAllester. 1999. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory, COLT 1999, Santa Cruz, CA, USA, July 7-9, 1999*, pages 164–170. ACM.

OpenAI. 2023. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt. https://openai.com/blog/chatgpt.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2814–2828. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4498–4507. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pages 1–5. IEEE.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509. The Association for Computational Linguistics.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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 *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1737–1748. ACM / IW3C2.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4281–4294. Association for Computational Linguistics.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 950–958. ACM.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Yanbin Wei, Qiushi Huang, James T. Kwok, and Yu Zhang. 2024. KICGPT: large language model with knowledge in context for knowledge graph completion. *CoRR*, abs/2402.02389.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion of The Web Conference 2022, Virtual Event / Lyon, France, April 25 - 29, 2022*, pages 162–165. ACM.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2319–2328.

Rui Yang, Jiahao Zhu, Jianping Man, Hongze Liu, Li Fang, and Yi Zhou. 2025. GS-KGC: A generative subgraph-based framework for knowledge graph completion with large language models. *Inf. Fusion*, 117:102868.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29476–29490.

A Theoretical Justification

A.1 Complexity Analysis of Pseudo-Neighbor Retrieval

In our SGNE module, we retrieve the top-k_s *structural pseudo-neighbors* for each query or candidate entity by computing cosine similarity in the embedding space. Let the number of total entities be **N**, the embedding dimension be **d**, and the number of queries be **Q**. Then, for each query, the brute-force computation of cosine similarity has complexity:

$$\mathcal{O}(N \cdot d). \tag{15}$$

Thus, the total retrieval cost becomes:

$$\mathcal{O}(Q \cdot N \cdot d). \tag{16}$$

To reduce this cost in large-scale knowledge graphs, approximate nearest neighbor (ANN) methods such as FAISS can be used, reducing the retrieval complexity to:

$$\mathcal{O}(Q \cdot \log N \cdot d). \tag{17}$$

This optimization enables scalable retrieval even for entity sets with millions of entries, as shown in (Johnson et al., 2021). We further observe that the top-k_s neighbors are precomputed and cached during training, making the cost negligible at inference time.

A.2 Computational Cost Analysis

To assess the computational efficiency of **SLiNT**, we compare it against two representative baselines on the FB15k-237 dataset: (i) a vanilla **LLaMA** model without structural prompts or tuning, and (ii) **DIFT**, which incorporates structural embeddings via prompts but lacks our proposed modules (SGNE, DHCL, GDDI). We evaluate the training time per epoch and inference latency across three structural encoders (TransE, SimKGC, CoLE) on a single 64GB MetaX GPU. Table 6 summarizes the results. Although **SLiNT** integrates multiple modules—including pseudo-neighbor retrieval, attention-based fusion, contrastive learning, and token-level injection—it introduces only $\sim$3% additional training time and less than 4% inference latency overhead relative to the DIFT baseline. This efficiency is achieved through several design choices: (1) pseudo-neighbor retrieval is cached and reused during training; (2) the LLaMA backbone is kept frozen; and (3) LoRA adapters are lightweight and efficient. These results demonstrate that **SLiNT** is scalable and deployment-friendly without sacrificing computational efficiency.

A.3 Robustness and Generalization of Contrastive Loss

To support fine-grained structural discrimination, our DHCL module optimizes a contrastive loss $\mathcal{L}_{\text{CL}}$, as defined in Section 3.2. This loss directly encodes structure-aware decision boundaries by comparing the distances between the query and interpolated hard positives/negatives in the structure space.

Stability via Lipschitz Continuity. Let $Z_j = \|\tilde{\mathbf{q}} - \tilde{\mathbf{n}}_j\|_2 - \|\tilde{\mathbf{q}} - \tilde{\mathbf{p}}_j\|_2$ denote the contrastive margin. Under unit-norm embeddings and interpolation sampling, we have $Z_j \in [-2, 2]$, and the per-term loss $\ell(Z_j) = -\log \sigma(Z_j)$ is 1-Lipschitz and bounded:

$$\ell(Z_j) \in [\log(1{+}e^{-2}),\ \log(1{+}e^{2})] \approx [0.13,\ 2.13].$$

This boundedness ensures stable gradients and robustness, especially when interpolated negatives lie close to the decision boundary.

PAC-Bayes Generalization Bound. We adopt a PAC-Bayes analysis (McAllester, 1999; Saunshi et al., 2019) to study generalization under structure-sensitive contrastive training. Let the encoder f_θ be parameterized by $\theta \in \mathbb{R}^{d'}$, and assume a Gaussian prior $P = \mathcal{N}(0, \sigma^2 I)$ and posterior $Q = \mathcal{N}(\theta, \sigma^2 I)$. Then the expected risk satisfies:

$$\mathcal{R}(Q) \leq \hat{\mathcal{R}}(Q) + \sqrt{\frac{1}{2N}\left(\mathrm{KL}(Q\|P) + \log\frac{1}{\delta}\right)}, \tag{18}$$

with KL divergence given by:

$$\mathrm{KL}(Q\|P) = \frac{1}{2\sigma^2}\|\theta\|^2. \tag{19}$$

Because the contrastive loss $\mathcal{L}_{\text{CL}}$ is Lipschitz and bounded, it satisfies the assumptions of the PAC-Bayes framework. This result confirms that contrastive training under DHCL maintains stable generalization behavior even when interpolated negatives lie near structural decision boundaries.

A.4 Modal Alignment Theory for Structure Injection

Our GDDI implements structure injection via *token-level injection*, where the token embeddings of [QUERY] and top-k_r [ENTITY] markers are replaced with structure-enhanced vectors $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{e}}_i$. These embeddings are injected into a frozen LLM, forming the input:

$$\mathbf{X}_{\text{input}} = [\text{CLS},\ \tilde{\mathbf{q}},\ \ldots,\ \tilde{\mathbf{e}}_1,\ \ldots,\ \tilde{\mathbf{e}}_{k_r},\ \text{EOS}]. \tag{20}$$

We treat this process as a cross-modal alignment between graph-structured embeddings and language model token representations. Let the LLM be viewed as a conditional language model:

$$p_\theta(\mathbf{y}|\mathbf{X}). \tag{21}$$

The structural embedding injection aims to preserve the semantic consistency between $\mathbf{X}_{\text{struct}}$ (the

Encoder	Training Time (s/epoch)			Overhead	Inference Latency (ms/sample)		
	LLaMA	DIFT	SLiNT	vs. DIFT	LLaMA	DIFT	SLiNT
TransE	212.6	220.4	227.0	+3.0%	9.3	9.7	10.1
SimKGC	226.2	233.7	241.0	+3.1%	9.7	10.1	10.5
CoLE	231.5	239.3	246.6	+3.1%	10.0	10.4	10.8

Table 6: Training and inference cost of SLiNT vs. DIFT and LLaMA on FB15k-237.

injected representation) and the output $\mathbf{y}$. Under the information bottleneck (IB) principle (Tishby and Zaslavsky, 2015), we define the learning objective as:

$$\max I(\mathbf{X}_{\text{struct}}; \mathbf{y}) - \beta I(\mathbf{X}_{\text{struct}}; \mathbf{Z}). \quad (22)$$

where $\mathbf{Z}$ is the latent representation inside the LLM. This objective seeks a trade-off: inject structure such that it influences generation (high mutual info with $\mathbf{y}$), while not deviating excessively from LLM's internal representations.

In practice, we approximate this by replacing the token embeddings at designated slots with $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{e}}_i$, and minimizing the KL divergence between pre- and post-injection logits:

$$\mathcal{L}_{\text{align}} = \text{KL}(p_{\text{LM}}(\cdot|\mathbf{X}) \| p_{\text{LM}}(\cdot|\mathbf{X}_{\text{inject}}). \quad (23)$$

This provides a differentiable surrogate for modal alignment. If the structure injection preserves or improves generation quality, we can conclude successful alignment.

B Dataset Overview

We conduct experiments on two widely used link prediction benchmarks: FB15k-237 and WN18RR. Table 7 summarizes the key dataset statistics.

- **FB15k-237** is a refined subset of Freebase. It covers diverse entity types and relational patterns (e.g., 1-to-N, N-to-1). Redundant inverse edges are removed to avoid test leakage (Toutanova et al., 2015).
- **WN18RR** is derived from WordNet, modeling lexical and hierarchical semantics such as hypernymy and derivation. Reversible relations are eliminated for robust evaluation (Dettmers et al., 2018).

C Training and Implementation Details

We provide detailed configuration settings for all components of **SLiNT**.

Dataset	#Ent.	#Rel.	Train	Valid	Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134

Table 7: Basic statistics of benchmark datasets.

Hardware and Framework. All experiments are conducted on 8×64GB MetaX GPUs, a domestic CUDA-compatible accelerator with performance comparable to NVIDIA A100s. Our implementation is based on PyTorch with automatic mixed-precision (AMP) training for efficiency.

Backbone Model. We use the frozen LLaMA-2-7B model from HuggingFace[2] as the base language model. Structure-aware features are injected via prompt and token-level replacements without updating LLM parameters. LoRA is used for efficient adaptation, with rank $r = 128$, scaling factor $\alpha = 64$, and dropout rate of 0.1.

KG Embeddings. We experiment with three pretrained KG encoders: TransE, SimKGC, and CoLE. Each query is used to retrieve a top-m candidate list from a pretrained KG embedding model, with $m = 20$. These embeddings provide the structural foundation for neighborhood enhancement and contrastive supervision.

SGNE Settings. In the SGNE module, we retrieve top-$k_s = 5$ pseudo-neighbors for each query or candidate entity using cosine similarity in the KG embedding space. The query and its neighbors are fused with multi-head attention, and the outputs are cached for efficiency.

DHCL Settings. For contrastive training, we sample $N = 50$ candidate entities per query. The contrastive loss is computed over $k_c = 10$ hard positives and negatives selected based on pseudo-neighbor overlap. The loss is weighted by a confusion-aware scoring function. The contrastive loss coefficient is $\lambda = 0.5$.

[2] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

GDDI Settings. In the GDDI module, we inject $k_r = 1$ structure-enhanced entity token into each input sequence. The enhanced embeddings replace the placeholders for `[QUERY]` and `[ENTITY]` tokens. Injection is performed at both prompt-level (text) and token-level (embedding) positions.

Optimization and Training. All models are trained using the Adam optimizer with a learning rate of 2×10^{-5} and a batch size of 64. We train for 3 epochs with early stopping based on validation MRR. Random seeds are fixed for reproducibility, and all results are averaged over three runs.

D Case Study

To demonstrate how SLiNT leverages pseudo-neighbor injection and contrastive learning, we present three representative cases from WN18RR and FB15k-237. Each case includes a query, candidates, SGNE-retrieved pseudo-neighbors, and model predictions. Tables 8–10 are presented alongside their respective discussions.

D.1 Case 1: Disambiguating Musical Components

Query	(instrument, has_part, ?)
Candidates	{bow, string, keyboard, bridge}
Ground Truth	string
Pseudo-Neighbors (Top-5)	violin, cello, harp, guitar, banjo
LLaMA + CoLE	keyboard (Top-1)
DIFT + CoLE	keyboard (Top-1)
SLiNT + CoLE	**string** (Top-1)

Table 8: Case 1: SLiNT correctly predicts `string` by leveraging structurally similar instruments.

Analysis. LLaMA and DIFT choose `keyboard`, a plausible but structurally irrelevant part of KG. SLiNT identifies `string` by leveraging pseudo-neighbors such as `violin` and `cello`, where `string` is a shared component.

D.2 Case 2: Differentiating Geopolitical Containment

Query	(?, location_contains, mountain)
Candidates	{Nepal, Asia, Everest, Tibet}
Ground Truth	Nepal
Pseudo-Neighbors (Top-5)	Himalaya, Kathmandu, Pokhara, Lumbini, Mustang
LLaMA + CoLE	Asia (Top-1)
DIFT + CoLE	**Nepal** (Top-1)
SLiNT + CoLE	**Nepal** (Top-1)

Table 9: Case 2: SLiNT correctly identifies `Nepal` by grounding in local structural cues.

Analysis. Although all candidates are semantically relevant to `mountain`, SLiNT uses structure-guided cues, e.g., `Himalaya` and `Kathmandu` to localize the correct geopolitical scope.

D.3 Case 3: Failure Case on Long-tail Relation

Query	(person, known_for, ?)
Candidates	{acting, painting, novel, photography}
Ground Truth	painting
Pseudo-Neighbors (Top-5)	artist, sculptor, painter, curator, illustrator
LLaMA + CoLE	acting (Top-1)
DIFT + CoLE	acting (Top-1)
SLiNT + CoLE	novel (Top-1)

Table 10: Case 3: SLiNT fails to predict `painting`, despite partial structural grounding.

Analysis. All models fail to predict `painting`, revealing challenges in handling long-tail relations. SLiNT ranks `novel` highest, likely influenced by relevant creative-profession neighbors, yet fails to fully disambiguate the semantic role of the candidate. This failure highlights the limitations of structure-based grounding in the absence of sufficient semantic alignment.