GLTW: Joint Improved Graph Transformer and LLM via Three-Word Language for Knowledge Graph Completion

Anonymous ACL submission

Abstract

Knowledge Graph Completion (KGC), which aims to infer missing or incomplete facts, is a crucial task for KGs. However, integrating the vital structural information of KGs into Large Language Models (LLMs) and outputting predictions deterministically remains challenging. To address this, we propose a new method called **GLTW**, which encodes the structural information of KGs and merges it with LLMs to enhance KGC performance. Specifically, we introduce an improved Graph Transformer (**iGT**) that effectively encodes subgraphs with both local and global structural information and inherits the characteristics of language model, bypassing training from scratch. Also, we develop a subgraph-based multi-classification training objective, using all entities within KG as classification objects, to boost learning efficiency. Importantly, we combine iGT with an LLM that takes KG language prompts as input. Our extensive experiments on various KG datasets show that GLTW achieves significant performance gains compared to SOTA baselines.

1 Introduction

Knowledge Graphs (KGs) are pivotal resource for a multitude of knowledge-intensive intelligent tasks (e.g., question answering (Zhai et al., 2024), recommendation systems (Zhao et al., 2024), planning (Wang et al., 2024), and reasoning (Chen et al., 2024b), among others). They are composed of a vast number of triplets in the format of (h, r, t), where h and t represent the head and tail entities, respectively, and r denotes the relationship connecting these two entities. However, popular existing KGs, such as Freebase (Bollacker et al., 2008), WordNet (Miller, 1995), and WikiData (Vrandečić and Krötzsch, 2014), suffer from a significant drawback: the presence of numerous incomplete or missing triplets, thereby giving rise to the task of KG Completion (KGC). KGC aims to accurately predict the missing triplets by leveraging known entities and relations for effectively enhancing KGs.

In recent years, with super-sized training corpora and computational cluster resources, Large Language Models (LLMs) have developed rapidly and enabled state-of-the-art performance in a wide range of natural language tasks (Touvron et al., 2023; Qin et al., 2023; Liu et al., 2024a). Consequently, certain studies have applied LLMs to KGC tasks. For instance, (Yao et al., 2023; Zhu et al., 2024; Wei et al., 2024) utilize zero/few-shot In-Context Learning (ICL) to accomplish KGC, while (Li et al., 2024a; Xu et al., 2024) leverage LLMs to enhance the descriptions of entities and relations in KGs, thereby improving text-based KGC methods (Yao et al., 2019; Zhang et al., 2020b; Wang et al., 2022b; Liu et al., 2022; Wang et al., 2022c; Yang et al., 2024a). Intuitively, integrating non-textual structured information appropriately can augment LLMs' understanding and representation of KGs. For example, (Zhang et al., 2024; Liu et al., 2024b; Guo et al., 2024) combine graph-structured information with LLMs to boost KGC tasks.

Yet, they either use traditional embedding-based KGC methods (Bordes et al., 2013; Lin et al., 2015; Sun et al., 2019; Balažević et al., 2019) that only consider internal links of triplets or rely on Graph Neural Networks (GNNs) (Bronstein et al., 2021; Corso et al., 2020) that merely encode local subgraphs, thus both missing out on global structural knowledge. Also, LLMs, typically used for generative tasks, have long been troubled by hallucination (Ji et al., 2023; Rawte et al., 2023). In contrast, the prediction targets of KGC are generally confined to the given KG, making it unwise to directly integrate LLMs into KGC tasks[1]. In short, how to encode both local and global structural information of KGs and combine it with knowledge-rich LLMs to achieve deterministic KGC remains un-

[1]Notably, see Appendix A for more related works.

derexplored.

To this end, we propose a novel method (named **GLTW**), which effectively encodes KG subgraphs with both local and global structural information and integrates LLMs in a deterministic fashion to improve the performance of KGC. Concretely, we first treat entities and relations within KG as inseparable units, adding them as tokens to the original Tokenizer, while referring to triplets as three-word sentences (Guo et al., 2024). Subsequently, for each target triple, we extract a subgraph that encompasses both local and global structural information from the given training KG data (Section 3.1). To effectively process the subgraph, we introduce an improved Graph Transformer (**iGT**), which takes the entity and relation embeddings (initialized by a pooling operation), the relative distance matrix, and the relative distinction matrix of the subgraph as inputs, and encodes them using the enhanced graph attention mechanism (Section 3.2). Furthermore, we construct multiple positive and negative triplet samples from the subgraph, which are used to build the subgraph-based multi-classification training objective with all entities within the KG as classification objects (Section 3.3). Finally, we merge iGT with an LLM that takes KG language prompt as input (Section 3.4). To sum up, we highlight our contributions as follows:

- We formulate a novel method, GLTW, which aims to encode both local and global structural information of KG and amalgamate it with LLMs to enhance KGC performance. Note that we consider KGC as a subgraph-based multi-classification task, outputting prediction probabilities for all entities from KG at once.
- We introduce iGT, which simplifies the complexity of positional encoding for subgraphs, enlarges the size of subgraphs, and treats entities and relations in a differentiated yet fair manner. Importantly, it inherits the characteristics of language model, thereby avoiding training from scratch.
- We conduct extensive experiments on three commonly used KG datasets (i.e., WN18RR, FB15k-237, and Wikidata5M) to show that GLTW is highly competitive compared with other state-of-the-art baselines. Meanwhile, ablation studies demonstrate the efficacy and indispensability for core modules and key parameters.

2 Preliminaries

2.1 Task Definition

Knowledge graphs (KGs) are directed graphs that can be formally represented as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote respectively the sets of entities and relations, and $\mathcal{T} = \{(h, r, t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ defines a collection of triples. The goal of KGC is to accurately predict the incomplete triples that exist within $\mathcal{G}$. In this paper, we focus on the **link prediction** task, a key component of KGC. This task is designed to predict the missing entity ? in a given triple $(h, r, ?)$ or $(?, r, t)$. We unify the link prediction task into tail entity prediction by constructing inverse relation $r^{-1} \in \mathcal{R}^{-1}$, i.e., $(t, r^{-1}, ?)$.

2.2 Graph Transformer

The attention mechanism (Shehzad et al., 2024) in a graph transformer can be expressed as follows:

$$\operatorname{softmax}\left(\frac{QK^\top}{\sqrt{d}} + B_P + M\right)V, \qquad (1)$$

where Q, K, and V denote the query, key and value matrices, and d represents the query and key dimension. The matrices B_P and M serve the purposes of Positional Encoding (PE) and masking. In GLM (Plenz and Frank, 2024), $B_P = f(P)$, where P is the relative distance matrix based on Levi graph of subgraph (as shown in Fig. 5(a)-(b) in Appendix C), and f is an element-wise function; M is a zero matrix[2]. This non-invasive modification avoids pre-training from scratch and preserves compatibility with the language model parameters.

2.3 Three-word Language

The concept of the three-word language originates from the MKGL method proposed by (Guo et al., 2024), which considers individual entities and relations as indivisible tokens and incorporates them into the LLM tokenizer (i.e., expanded tokenizer). For example, entity *black poodle* and relation *is a* are encoded as tokens <*kgl: black poodle*> and <*kgl: is a*>, respectively, and are employed to construct corresponding KG language prompt (see Appendix D). To prevent training these new tokens from scratch, MKGL utilizes a GNN encoder to derive their embeddings from the original tokenizer based on the textual and structural information of the entities/relations. This enables LLMs to effectively navigate and master the three-word language.

[2] In this paper, we focus on the global GLM (gGLM), which invokes an additional G2G relative position to access distant triplets and sets M is a zero matrix.

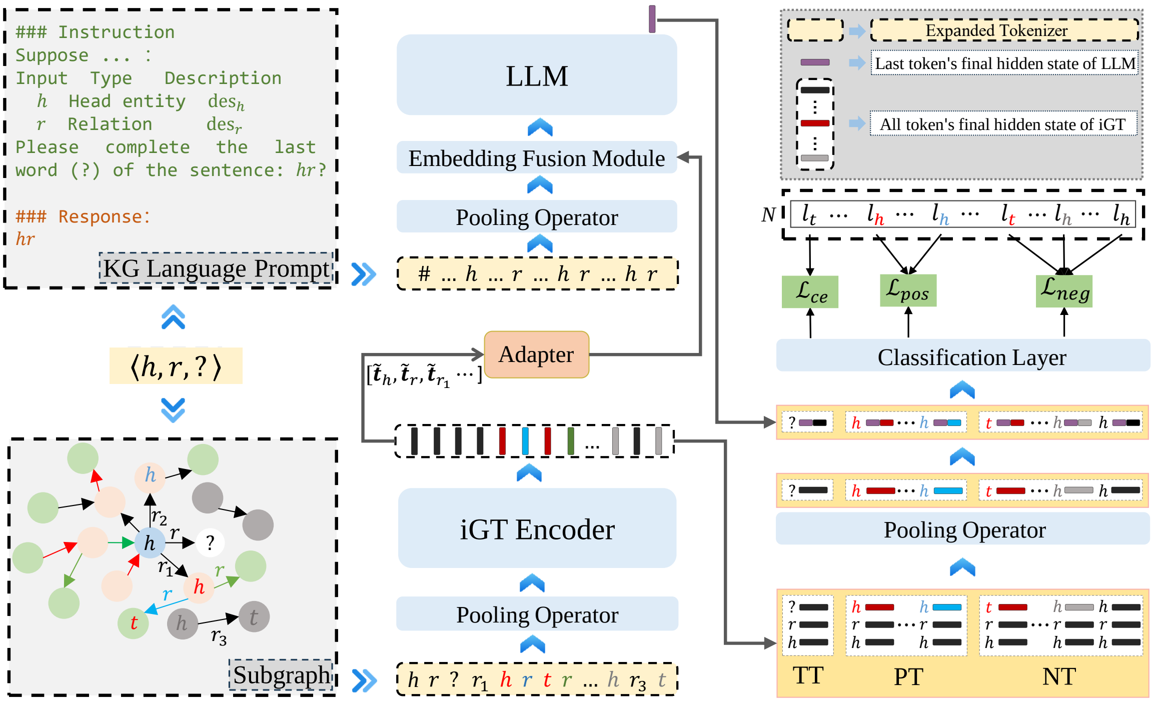


Figure 1: The pipeline of GLTW. $\mathcal{L}_{ce}$, $\mathcal{L}_{pos}$, and $\mathcal{L}_{neg}$ are loss objectives for Target Triplet (TT), Positive Triplets (PT) and Negative Triplets (NT), respectively. Notably, the r, r_1, r_2, and r_3 highlighted in black pertain to the same relation but exist in different triplets. For simplicity, h and h can be either head or tail entities, as they are shared by multiple triplets.

3 Method

In this section, we elaborate on our proposed method, GLTW, in four parts: *Subgraph Extraction*, *Improved Graph Transformer*, *Subgraph-based Training Objective*, and *Joint iGT and LLM*. Figure 1 illustrates the pipeline of GLTW. Notably, the KG language prompt in this paper directly follows that of MKGL (Guo et al., 2024).

3.1 Subgraph Extraction

Before training or prediction, we extract a subgraph $\mathcal{G}_{sub}(h,r,t)$ for each target triplet (h,r,t) from $\mathcal{G}$. For consistent training and prediction, we require that the subgraph only comprises triplets sampled from given h and r, represented as $\mathcal{G}_{sub}(h,r,?)$. $\mathcal{G}_{sub}(h,r,?)$ contains three types of triplet subsets: T_{hr}, T_h and T_r, where T_{hr} and T_h hold neighboring triplets around $(h,r,?)$, and T_r samples distant (global) triplets with r. For T_{hr} and T_h, we set the sampling radius as l, then $T_{hr/h} = \cup_{i=1}^{l} T_{hr/h}^{i}$. Specifically, when $l=1$, $T_{hr}^1 = \{(h,r,t^1)|t^1 \in \mathcal{E}-\{t\}\}$ and $T_h^1 = \{(h,r^1,t^1)/(t^1,r^1,h)|r^1 \in \mathcal{R} -\{r\}, t^1 \in \mathcal{E}\}$; when $l>1$, $T_{hr/h}^i = \{(h^{i-1},r^i,t^i)/(t^i,r^i,h^{i-1})|h^{i-1} \in New(T_{hr/h}^{i-1}), r^i \in \mathcal{R}, t^i \in \mathcal{E}\}$, where "/" denotes "or", and $New(T_{hr/h}^{i-1})$ is the latest sampled entity set in $T_{hr/h}^{i-1}$. For T_r, we solely consider distant triplets with r, i.e., $T_r = \{(h',r,t')|h',t' \in \mathcal{E}-\{h,t\}\}$.

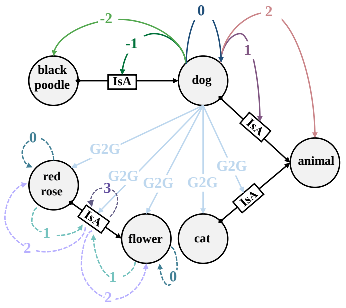
In the sampling process (e.g., $T_{hr/h}^i$ and T_r), we leverage Random Walk (Ko et al., 2024) to select triplets based on the degree distribution of candidate entities, considering both out-degree and in-degree. Additionally, to control the size of the subgraph, we set the total number of sampled triplets to $m = m_{hr} + m_h + m_r$, where $m_{hr/h}$ and m_r represent the sampling numbers of $T_{hr/h}$ and T_r, respectively. Note that if $|T_{hr/h}| < m_{hr/h}$, we select more distant triplets to ensure m.

3.2 Improved Graph Transformer

In order to effectively encode $\mathcal{G}_{sub}(h,r,?)$, we propose an improved Graph Transformer (iGT). Concretely, we first introduce the three-word language and pre-compress the textual information of entities and relations. Given an entity e and a relation r from $\mathcal{G}_{sub}(h,r,?)$, their token embedding sequences of textual information take the following forms:

$$\mathrm{E}_e = [\boldsymbol{t}_e^1, \cdots, \boldsymbol{t}_e^{n_e}], \mathrm{E}_r = [\boldsymbol{t}_r^1, \cdots, \boldsymbol{t}_r^{n_r}], \quad (2)$$

where n_e and n_r represent the lengths of the token sequences for textual information. Then, following (Guo et al., 2024), we draw on the pooling



(a) Example of subgraph with relative distances for *dog* and distinction for <*red rose, IsA, flower*>

	black poodle	is a	dog	is a	animal	cat	is a	red rose	is a	flower
black poodle	0	1	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is a	-1	0	1	G2G	G2G	G2G	G2G	G2G	G2G	G2G
dog	-2	-1	0	1	2	G2G	G2G	G2G	G2G	G2G
is a	G2G	G2G	-1	0	1	G2G	G2G	G2G	G2G	G2G
animal	G2G	G2G	-2	-1	0	-2	-1	G2G	G2G	G2G
cat	G2G	G2G	G2G	G2G	2	0	1	G2G	G2G	G2G
is a	G2G	G2G	G2G	G2G	1	-1	0	G2G	G2G	G2G
red rose	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	1	2
is a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-1	0	1
flower	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-2	-1	0

(b) Relative distance matrix P for (a)

	black poodle	is a	dog	is a	animal	cat	is a	red rose	is a	flower
black poodle	0	1	0	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is a	2	3	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G
dog	0	1	0	1	0	G2G	G2G	G2G	G2G	G2G
is a	G2G	G2G	2	3	2	G2G	G2G	G2G	G2G	G2G
animal	G2G	G2G	0	1	0	0	1	G2G	G2G	G2G
cat	G2G	G2G	G2G	G2G	0	0	1	G2G	G2G	G2G
is a	G2G	G2G	G2G	G2G	2	2	3	G2G	G2G	G2G
red rose	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	1	0
is a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	3	2
flower	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	1	0

(c) Relative distinction matrix D for (a)

Figure 2: Example of subgraph preprocessing in iGT. We follow the construction strategy of the relative position matrix P in gGLM (Plenz and Frank, 2024). The relative distinction matrix D differentiates entities and relations in iGT. Notably, it can be extended to gGLM, providing clear textual boundaries for entities and relations (see Appendix C). Also, entries with G2G are initialized to $+\infty$.

operator $\mathrm{Pool_{op}}()$ from PNA (Corso et al., 2020) to compress E_e and E_r, i.e.,

$$\boldsymbol{t}_e = \mathrm{Pool_{op}}(\mathrm{E}_e), \boldsymbol{t}_r = \mathrm{Pool_{op}}(\mathrm{E}_r), \quad (3)$$

where $\boldsymbol{t}_e$ and $\boldsymbol{t}_r$ denote the textual token embedding of e and r, respectively. By utilizing the pooling operator, we furnish embeddings for every entity and relation within $\mathcal{G}_{sub}(h, r, ?)$.

Next, we construct a relative distance matrix P with a global perspective for $\mathcal{G}_{sub}(h, r, ?)$, following GLM, as shown in Fig. 2 (a) and (b). We regard triplets as three-word sentences, where each token represents an entity or a relation, and calculate their relative distances. Moreover, the graph-to-graph (G2G) relative position (initialized as the parameter of the relative position for $+\infty$) can connect any token to other tokens, thereby enabling access to and learning of distant entities or relations.

Although P achieves graph manipulation in a non-intrusive way, it fails to distinguish between entities and relations in $\mathcal{G}_{sub}(h, r, ?)$, which may introduce confounding bias. This is because in KG, entities represent real-world objects or concepts, while relations describe the interactions between entities (Pan et al., 2024). To rectify this, we introduce a new relative distinction matrix D, which has the same shape as P and shares G2G, as shown in Fig. 2(c). Unlike P, D aims to distinguish between entities and relations in the subgraph. To be specific, the relative positions between entities (i.e., entity-entity) are set to 0 and populated into the corresponding ones in D. Similarly, the positions for entity-relation, relation-entity, and relation-relation pairs are assigned the values of 1, 2, and 3, respectively. Furthermore, we rewrite the Eq. (1) of the attention mechanism as:

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}} + B_{PD}\right)V, \quad (4)$$

where $B_{PD} = \frac{1}{2}\left(f_1(P) + f_2(D)\right)$. Here, f_1 and f_2 are two different element-wise functions. Compared with GLM, iGT focuses on the structural information of $\mathcal{G}_{sub}(h, r, ?)$, bringing several benefits: it simplifies the complexity of positional encoding; handles larger subgraphs; and differentiates between entities and relations while treating them equitably. Importantly, iGT inherits GLM's non-invasive properties, circumventing the need to train the model from scratch, although the pooling operator may lose some textual information.

With iGT, we can encode the subgraph $\mathcal{G}_{sub}(h, r, ?)$, and the overall process is as follows:

$$[h, r, ?, \cdots] = \mathrm{ExTok}\left(\mathcal{G}_{sub}(h, r, ?)\right),$$
$$[\boldsymbol{t}_h, \boldsymbol{t}_r, \boldsymbol{t}_?, \cdots] = \mathrm{Pool_{op}}\left(\mathrm{Emb}([h, r, ?, \cdots])\right),$$
$$[\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_?, \cdots] = \mathrm{iGT}([\boldsymbol{t}_h, \boldsymbol{t}_r, \boldsymbol{t}_?, \cdots], P, D),$$

where ExTok is the Expanded Tokenizer, which integrates entities and relations as new tokens into the existing vocabulary. Emb denotes Embedding layer. Of note, during training and prediction, we replace ? (to be predicted) with *mask* token from the original Tokenizer.

3.3 Subgraph-based Training Objective

In this paper, we frame the $(h, r, ?)$ prediction task as a multi-classification problem. To elaborate, we implement an MLP-based classification layer that takes $[\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_?]$ from iGT's final hidden layer as input, with its output dimension corresponding to the KG's total entity count N. Then, we compute classification probabilities through softmax activation and optimize using cross-entropy loss. Of note, prior to classification, we perform the pooling operation on $[\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_?]$. The process can be formulated as:

$$\tilde{\boldsymbol{t}}_{(h,r,?)} = \mathrm{Pool}_{\mathrm{op}}([\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_?]), \quad (5)$$

$$\hat{\boldsymbol{t}}_{(h,r,?)} = \mathrm{softmax}(\mathrm{MLP}(\tilde{\boldsymbol{t}}_{(h,r,?)})), \quad (6)$$

$$\mathcal{L}_{ce} = -\log(\hat{\boldsymbol{t}}_{(h,r,?),l_t}), \quad (7)$$

where $\hat{\boldsymbol{t}}_{(h,r,?),l_t}$ denotes the likelihood of entity t being selected.

We don't use $\tilde{\boldsymbol{t}}_?$ alone as the classification input; instead, we opt for $[\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_?]$. This is because iGT encodes $\mathcal{G}_{sub}(h, r, ?)$, in which h may be shared by multiple triplets, and r can also appear in several triplets (see Fig. 1). Thus, the optimization objective based solely on $\tilde{\boldsymbol{t}}_?$ may not effectively address the prediction task $(h, r, ?)$. Also, according to Section 3.1, triplets in T^1_{hr} feature the same head entity and relation as $(h, r, ?)$, such as (h, r_1, h) and (h, r_2, h) (see Fig. 1). Hence, during prediction, h and h can emerge as potential optimization targets requiring positive attention, whereas other entities, including h, warrant negative attention. To this end, we partition all entities in $\mathcal{G}_{sub}(h, r, ?)$ (excluding ?) into two sets: Pos and Neg. Pos includes tail entities from all triplets in T^1_{hr}, while Neg comprises the remaining entities. The optimization objectives for Pos and Neg take the following forms:

$$\mathcal{L}_{pos} = -\frac{1}{|\mathrm{Pos}|} \sum_{t' \in \mathrm{Pos}} \log(\hat{\boldsymbol{t}}_{(h,r,t'),l_{t'}}), \quad (8)$$

$$\mathcal{L}_{neg} = -\frac{1}{|\mathrm{Neg}|} \sum_{t' \in \mathrm{Neg}} \log(\hat{\boldsymbol{t}}_{(h,r,t'),l_{t'}}), \quad (9)$$

where $|\mathrm{Pos}|$ and $|\mathrm{Neg}|$ denote the number of entities in Pos and Neg.

Combining $\mathcal{L}_{ce}$, $\mathcal{L}_{pos}$, and $\mathcal{L}_{neg}$, the subgraph-based overall objective can be formalized as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + \beta_1(\mathcal{L}_{pos} - \beta_2 \mathcal{L}_{neg}), \quad (10)$$

where $\beta_1 > 0$ and $\beta_2 > 0$ are tunable hyperparameters. During training, to prevent $\mathcal{L}_{neg}$ from dominating excessively, we employ the following strategy to adjust β_2 adaptively:

$$\beta_2 = \begin{cases} 1, \mathcal{L}_{pos} > \mathcal{L}_{neg}, \\ 0.5 * \frac{\mathcal{L}_{pos}}{\mathcal{L}_{neg}}, \mathcal{L}_{pos} \leq \mathcal{L}_{neg}. \end{cases} \quad (11)$$

3.4 Joint iGT and LLM

We now combine iGT and LLM by fusing entity and relation embeddings. To be specific, we integrate the pooled embeddings of entity h ($\boldsymbol{t}^{llm}_h$) and relation r ($\boldsymbol{t}^{llm}_r$) from the LLM-based KG language prompt for $(h, r, ?)$ with iGT's output embeddings $[\tilde{\boldsymbol{t}}_h, \tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_{r_1}, \cdots]$, excluding $\tilde{\boldsymbol{t}}_?$. The process (i.e., the Embedding Fusion Module) is defined as:

$$\bar{\boldsymbol{t}}_r = \mathrm{Pool}_{\mathrm{op}}([\tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_{r_1}, \cdots]), \quad (12)$$

$$\boldsymbol{t}^{llm}_h \leftarrow (1-\lambda) \cdot \boldsymbol{t}^{llm}_h + \lambda \cdot \mathrm{Adapter}(\tilde{\boldsymbol{t}}_h), \quad (13)$$

$$\boldsymbol{t}^{llm}_r \leftarrow (1-\lambda) \cdot \boldsymbol{t}^{llm}_r + \lambda \cdot \mathrm{Adapter}(\bar{\boldsymbol{t}}_r), \quad (14)$$

where $\lambda \in [0, 1]$, and Adapter aims to align embedding dimensions. The selection of Adapter is flexible. In practice, following (Zhu et al., 2023), we implement Adapter as a simple projection layer. Notably, we pass the pooled relation embeddings $[\tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_{r_1}, \cdots]$ from $\mathcal{G}_{sub}(h, r, ?)$ to the LLM, enabling it to capture global KG structural information.

Then, we incorporate the embedding vector $\boldsymbol{t}^{llm}_{hr}$ from the last token of the LLM's final hidden layer into the classification layer as follows:

$$\tilde{\boldsymbol{t}}_{(h,r,?)} \leftarrow \mathrm{Concat}(\boldsymbol{t}^{llm}_{hr}, \tilde{\boldsymbol{t}}_{(h,r,?)}), \quad (15)$$

where $\tilde{\boldsymbol{t}}_{(h,r,?)}$ is derived from Eq. (5). Similarly, all positive and negative triplets constructed in Section 3.3 are combined with $\boldsymbol{t}^{llm}_{hr}$ in the same manner (see Fig. 1). Of note, the input dimension of the MLP classification layer changes accordingly through Eq. (15).

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate different methods on three widely used KG datasets, including FB15k-237 (Toutanova et al., 2015), WN18RR (Dettmers et al., 2018), and Wikidata5M (Vrandečić and Krötzsch, 2014), for the link prediction task. We detail these datasets in Table 4 from Appendix B.

Baselines. To assess the effectiveness of our methods, we follow (Plenz and Frank, 2024) by adopting the bidirectional encoder of T5-base as the base Pre-trained Language Model (PLM) for **iGT**. Meanwhile, we choose three LLMs with varying sizes for GLTW: Llama-3.2-1B/3B-Instruct (Dubey et al., 2024) and Llama-2-7b-chat (Touvron et al., 2023). For clarity, we denote GLTW with different LLMs as $\textbf{GLTW}_{1b/3b/7b}$.

Methods	FB15k-237				WN18RR				Wikidata5M			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	0.279	0.198	0.376	0.441	0.243	0.043	0.441	0.532	0.392	0.323	0.432	0.509
RotatE	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571	0.403	0.334	0.441	0.523
HAKE	0.346	0.250	0.381	0.542	0.497	0.452	0.516	0.582	0.394	0.322	0.435	0.521
CompoundE	0.350	0.262	0.390	0.547	0.492	0.452	0.510	0.570	-	-	-	-
KG-BERT	-	-	-	0.420	0.216	0.041	0.302	0.524	-	-	-	-
KG-S2S	0.336	0.257	0.373	0.498	0.574	0.531	0.595	0.661	-	-	-	-
CSProm-KG	0.358	0.269	0.393	0.538	0.575	0.522	<u>0.596</u>	<u>0.678</u>	0.380	0.343	0.399	0.446
PEMLM-F	0.355	0.264	0.389	0.538	0.556	0.509	0.573	0.648	-	-	-	-
CompGCN	0.355	0.264	0.390	0.535	0.479	0.443	0.494	0.546	-	-	-	-
REP-OTE	0.354	0.262	0.388	0.540	0.488	0.439	0.505	0.588	-	-	-	-
KRACL	0.360	0.266	0.395	0.548	0.527	0.482	0.547	0.613	-	-	-	-
gGLM	0.321	0.241	0.342	0.486	0.290	0.304	0.395	0.487	-	-	-	-
iGT (**ours**)	0.364	0.283	0.411	0.566	0.534	0.496	0.536	0.617	0.397	0.342	0.428	0.526
GPT-3.5	-	0.267	-	-	-	0.212	-	-	-	-	-	-
Llama-2-13B	-	-	-	-	-	0.315	-	-	-	-	-	-
KICGPT	0.412	0.327	0.448	0.554	0.549	0.474	0.585	0.641	-	-	-	-
MPIKGC-S	0.359	0.267	0.395	0.543	0.549	0.497	0.568	0.652	-	-	-	-
KG-FIT	0.362	0.275	-	0.572	-	-	-	-	-	-	-	-
MKGL	0.415	0.325	0.454	0.591	0.552	0.500	0.577	0.656	-	-	-	-
GLTW$_{1b}$	0.385	0.312	0.427	0.578	0.549	0.514	0.558	0.645	0.405	0.356	0.452	0.531
GLTW$_{3b}$	<u>0.427</u>	<u>0.338</u>	<u>0.462</u>	<u>0.599</u>	<u>0.578</u>	<u>0.538</u>	0.593	0.676	<u>0.429</u>	<u>0.376</u>	<u>0.476</u>	<u>0.553</u>
GLTW$_{7b}$	**0.469**	**0.351**	**0.481**	**0.614**	**0.593**	**0.556**	**0.649**	**0.690**	**0.457**	**0.414**	**0.506**	**0.587**

Table 1: Performance comparison of various methods across different datasets. Note that **bold** indicates the overall best performance, while <u>underline</u> marks the second-best one.

Also, we compare GLTW and iGT against numerous embedding-based, text-based, GNN/GT-based and LLM-based baselines. The embedding-based baselines include TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), HAKE (Zhang et al., 2020a), and CompoundE (Ge et al., 2023). The text-based baselines encompass KG-BERT (Yao et al., 2019), KG-S2S (Chen et al., 2022), CSProm-KG (Chen et al., 2023), and PEMLM-F (Qiu et al., 2024). The GNN/GT-based baselines cover CompGCN (Vashishth et al., 2019), REP-OTE (Wang et al., 2022a), and KRACL (Tan et al., 2023) (based on GNN), as well as gGLM (Plenz and Frank, 2024) (based on GT). Note that gGLM and iGT are trained on identical subgraphs. The LLM-based baselines comprise GPT-3.5-Turbo with one-shot ICL (marked as GPT-3.5) (Zhu et al., 2024), Llama-2-13B+Struct (marked as Llama-2-13B) (Yao et al., 2023), KICGPT (Wei et al., 2024), MPIKGC-S (Xu et al., 2024), KG-FIT (Jiang et al., 2024), and MKGL (Guo et al., 2024).

Configurations. In all experiments, unless otherwise specified, we default to setting $l = 2$ and $\overline{m} = m_{hr} = m_h = m_r = m/3 = 5$ for subgraph sampling. Meanwhile, we set $\lambda = 0.5$ and $\beta_1 = 0.5$. Of note, β_2 is adaptively calculated based on Eq. (11). Also, we assess performance by leveraging the Mean Reciprocal Rank (MRR) of target entities and the percentage of target entities ranked in the top k ($k = 1, 3, 10$), referred to as Hits@k. Due to space limitations, the complete experimental settings are provided in Appendix B.

4.2 Results Comparison

We compare the proposed methods with various KGC baselines on FB15k-237, WN18RR, and Wikidata5M, with the results shown in Table 1. The results indicate that: 1) GLTW$_{7b}$ consistently outperforms all competitors across all metrics, achieving overall gains of 8.5% in MRR, 6.9% in Hits@1, 10.2% in Hits@3, and 6.1% in Hits@10 compared to the second-best results (mostly from GLTW$_{3b}$). Meanwhile, GLTW's performance improves as the LLM size increases. These results demonstrate that GLTW effectively captures the characteristics of entities and relations in KGs and leverages the rich knowledge in LLMs to enhance prediction accuracy. 2) GLTW$_{3b}$ beats Llama-2-7b-based baseline MKGL (the most comparable method) on all metrics for FB15k-237 and WN18RR, with further improvements achieved by GLTW$_{7b}$. We attribute GLTW's advantage to its effective encoding of both local and global structural information of KGs, tailoring a suitable objective function for training subgraphs, and enabling LLMs to perceive structural information and effectively participate in entity prediction. 3) The proposed iGT consistently outstrips other GT/GNN-based baselines on FB15k-237 and WN18RR, while gGLM uniformly lags behind others. A detailed analysis is provided in the Ablation Study (see Section 4.3).

4.3 Ablation Study

In this section, we carefully demonstrate the efficacy and indispensability of the core modules and key parameters in our methods on FB15k-237 and WN18RR.

Method	FB15k-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
GLTW$_{1b}$	**0.385**	**0.312**	**0.427**	**0.578**	**0.549**	**0.514**	**0.558**	**0.645**
-w/o. iGT	0.108	0.082	0.177	0.303	0.205	0.157	0.261	0.413
-w/o. FT for LLM	0.379	0.291	0.397	0.572	0.539	0.509	0.545	0.629
GLTW$_{3b}$	**0.427**	**0.338**	**0.462**	**0.599**	**0.578**	**0.538**	**0.593**	**0.676**
-w/o. iGT	0.171	0.145	0.191	0.325	0.287	0.222	0.309	0.439
-w/o. FT for LLM	0.411	0.323	0.445	0.587	0.552	0.529	0.567	0.663
GLTW$_{7b}$	**0.469**	**0.351**	**0.481**	**0.614**	**0.593**	**0.556**	**0.649**	**0.690**
-w/o. iGT	0.207	0.184	0.236	0.366	0.394	0.309	0.357	0.462
-w/o. FT for LLM	0.438	0.343	0.465	0.607	0.568	0.538	0.612	0.677
-w/o LLM (i.e., iGT)	0.364	0.283	0.411	0.566	0.534	0.496	0.536	0.617

Table 2: Impact of each component for GLTW.

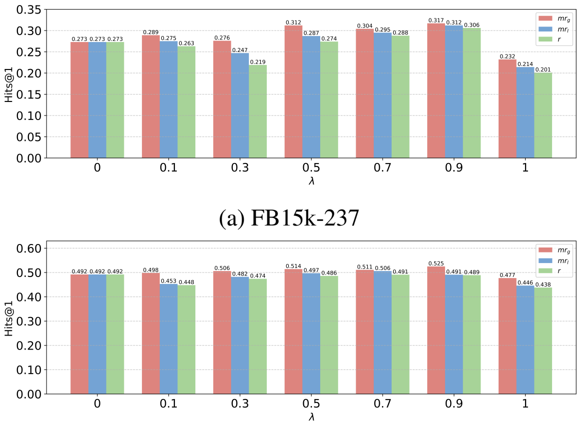
Necessity of each component for GLTW. To investigate the impact of iGT and LLMs on the performance for GLTW, we establish three control baselines: training iGT alone (w/o. LLM), fine-tuning LLMs alone (w/o. iGT), and using GLTW without fine-tuning LLMs (w/o. FT for LLM). For LLM fine-tuning alone, we input the KG language prompt and use the embedding vector of the last token from the final hidden layer as input to the classification layer. We report the results in Table 2. One can observe that iGT and LLMs exhibit significant performance drops compared to GLTW with different-sized LLMs. Specifically, iGT sees average declines of 5.1%, 4.5%, 5.5%, and 4.2% in MRR, Hits@1, Hits@3, and Hits@10, respectively, while LLMs experience average drops of 27.2%, 25.2%, 27.3%, and 24.9% in these metrics. Notably, GLTW without fine-tuning LLMs still surpasses both iGT and LLMs. This confirms that combining iGT and LLMs enhances entity prediction, consistent with prior works (Qiu et al., 2024; Zhang et al., 2024). Meanwhile, our proposed joint strategy effectively unlocks the LLM's potential for the link prediction task. Additionally, iGT consistently trumps all LLMs, underscoring the critical importance of relevant KG information and a well-designed training objective for performance improvements.

Utility of $\mathcal{L}$ and D. We delve into the subgraph-based training objective $\mathcal{L}$ and the relative discrimination matrix D by leveraging iGT. Thereafter, due to space limitations, we only report the values of Hits@1. For the **former**, we perform the leave-one-out test to explore the individual contributions of $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ to iGT, and further display the test results by simultaneously discarding them. As shown in Table 3, removing either $\mathcal{L}_{pos}$ or $\mathcal{L}_{neg}$ adversely affects the performance of iGT. In addition, the absence of both losses further worsens the decline of Hits@1, demonstrating that $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ are vital for training subgraphs. Interestingly, we observe that removing $\mathcal{L}_{pos}$ has a more pronounced negative impact than removing $\mathcal{L}_{neg}$. The empirical results indicate that in subgraph-based training, the construction of positive and negative triplets (i.e., PT and NT) is crucial for capturing structural information in KGs. For the **latter**, Table 3 reveals that removing D from iGT decreases the Hits@1 value by 2.9% and 4.3% on FB15k-237 and WN18RR, respectively. Importantly, extending D to gGLM improves Hits@1 value by 2.6% and 4.2% on these datasets. This suggests that B_{DP} enhances the relative positional encoding of entities and relations for subgraphs compared to B_P.

Method	FB15k-237				WN18RR			
	Hits@1	A.IT(↓)	A.IL(↓)	A.BBR(↓)	Hits@1	A.IT(↓)	A.IL(↓)	A.BBR(↓)
iGT	**0.283**	**16**	**38.43**	**0.00**	**0.496**	**16**	**44.16**	**0.00**
-w/o. $\mathcal{L}_{pos}$	0.263	-	-	-	0.471	-	-	-
-w/o. $\mathcal{L}_{neg}$	0.274	-	-	-	0.484	-	-	-
-w/o. $\mathcal{L}_{pos}$ & $\mathcal{L}_{neg}$	0.243	-	-	-	0.430	-	-	-
-w/o. D	0.254	-	-	-	0.453	-	-	-
gGLM	0.241	14.9	313.57	0.01	0.304	9.45	448.21	0.34
-w. D	0.267	-	-	-	0.346	-	-	-

Table 3: Utility of D and various parts in Eq. (10) for iGT, as well as iGT vs. gGLM

Notably, we illustrate the encoding strategy of D in gGLM, as shown in Fig. 5(c) of Appendix C. Essentially, D introduces boundaries to the textual descriptions of entities and relations in subgraphs, thereby augmenting the PLM's perception of triples within KG. Additionally, Table 3 records the three metrics during training for iGT and gGLM: average input triplets (**A.IT**), average input length (**A.IL**), and average tokens beyond the bucket range (**A.BBR**). The results show that iGT retains more input KG information than gGLM in terms of A.IT and A.BBR, especially on the WN18RR dataset. In contrast, the A.IL of gGLM is significantly higher than that of iGT, implying a higher computational cost for gGLM. Therefore, we speculate that gGLM's underperformance in the link prediction task may be due to: 1) the lack of clear boundaries for entities and relations; 2) significant information loss when handling KGs with lengthy textual descriptions; and 3) potential bias introduced by focusing more on entities or relations with longer textual descriptions in each triplet.



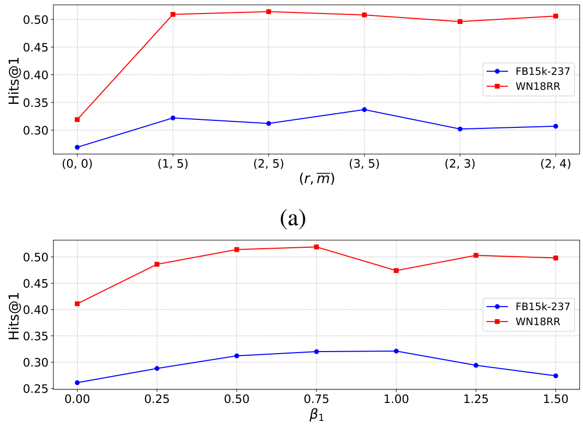
(a) FB15k-237

(b) WN18RR

Figure 3: Hits@1 with varying λ over FB15k-237 and WN18RR.

Varying λ. We explore the impacts of λ based on GLTW$_{1b}$ and select it from $\{0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Additionally, we compare the performance of various relation embeddings: those appearing in triplets from T_{hr} and T_h (i.e., $\bar{\boldsymbol{t}}_r = \mathrm{Pool}_{\mathrm{op}}([\tilde{\boldsymbol{t}}_r, \tilde{\boldsymbol{t}}_{r_1}, \cdots])$, marked as mr_l), those present in ones from T_{hr}, T_h and T_r (i.e., $\bar{\boldsymbol{t}}_r = \mathrm{Pool}_{\mathrm{op}}([\tilde{\boldsymbol{t}}_r, \cdots, \tilde{\boldsymbol{t}}_{r_3}, \cdots])$, marked as mr_g), and the single relation in the target triplet (i.e., $\bar{\boldsymbol{t}}_r = \tilde{\boldsymbol{t}}_r$, marked as r), as shown in Eq. (12). Fig. 3 shows that GLTW with $\lambda \notin \{0.0, 1.0\}$ consistently dominates that with $\lambda \in \{0.0, 1.0\}$ in terms of Hits@1. Moreover, the performance with $\lambda = 0$ is superior to that with $\lambda = 1$. Notably, the Hits@1 values for $mr_{l/g}$ and r are identical when $\lambda = 0$, as the LLM only takes the KG language prompt as input, independent of them. These results indicate that our proposed combination of iGT and LLM effectively improves link prediction. Furthermore, we find that both mr_g and mr_l consistently outperform r w.r.t. Hits@1, with mr_g uniformly surpassing mr_l. This demonstrates that incorporating local structural information (i.e., T_{hr} and T_h) from KG into the training process improves the prediction accuracy for target entities, while adding global structural information (i.e., T_r) further boosts performance significantly.

Varying $(r, \overline{m})$ and β_1. We look into the effects of the parameters$(r, \overline{m})$, which control subgraph shape, and the constraint parameter β_1 for $\mathcal{L}$ using GLTW$_{1b}$. First, we set $(r, \overline{m})$ to values in $\{(0, 0), (1, 5), (2, 5), (3, 5), (2, 3), (2, 4)\}$ and report the results in Fig. 4(a). Here, $(0, 0)$ means that the subgraph contains only the target triplet



(a)

(b)

Figure 4: Hits@1 with varying $(r, \overline{m})$ and β_1 over FB15k-237 and WN18RR.

$(h, r, ?)$. One can see that GLTW$_{1b}$ with $(r, m) = (0, 0)$ underperforms other cases w.r.t. Hits@1, indicating that incorporating graph structure information significantly enhances entity prediction. Furthermore, when $r = 2$, GLTW$_{1b}$'s Hits@1 value improves as $\overline{m}$ increases, suggesting that moderately enlarging the subgraph scale intensifies performance. However, when $\overline{m} = 5$, GLTW$_{1b}$'s performance does not monotonically improve with increasing r, highlighting the significantly impact of the subgraph sampling strategy on GLTW$_{1b}$'s performance for a given $\overline{m}$. For β_1, we select values from $\{0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$, as shown in Fig. 4(b). We observe that the Hits@1 score of GLTW$_{1b}$ initially rises and then declines as β_1 increases. This indicates that the optimal β_1 depends on the scenario and requires case-specific tuning.

5 Conclusion

In this paper, we propose a novel method, GLTW, which aims to encode the structural information of KGs and integrate it with LLMs to enhance KGC performance. Specifically, we formulate an improved graph transformer (iGT) that effectively encodes subgraphs with both local and global structural information and inherits the characteristics of language models, thus circumventing the need for training from scratch. Also, we develop a subgraph-based multi-classification training objective that treats all entities within KG as classification objects to improve learning efficiency. Importantly, we combine iGT with an LLM that takes KG language prompt as input. Finally, we conduct extensive experiments to verify the superiority of GLTW.

Limitations

Although empirical experiments have confirmed the effectiveness of the proposed GLTW, it still has two main limitations. **First**, training GLTW involves distinct original vocabularies for the T5 and Llama series, resulting in separate vocabularies for iGT and LLM. We speculate that well-trained GLTW on a unified vocabulary could further enhance its performance, but this would require training the models from scratch. **Second**, our proposed method uses pooling operations from PNA to compress textual information, which inevitably leads to some information loss. However, a key advantage of pooling operations is that they do not introduce new parameters requiring optimization. Even when training resources are limited and LoRA technology (Hu et al., 2021) is drawn to reduce memory consumption, the additional trainable parameters are negligible. Therefore, it is crucial to develop pooling operations that minimize such information loss, which we leave to future work.

Ethical Considerations

In this paper, all research and experiments utilize publicly available open-source datasets and models. We will release our code to support open research. Therefore, there is no ethical consideration in this paper.

References

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

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

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu. 2024a. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. *arXiv preprint arXiv:2209.07299*.

Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. *arXiv preprint arXiv:2307.01709*.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. 2024b. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *arXiv preprint arXiv:2410.23875*.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2020. Hitter: Hierarchical transformers for knowledge graph embeddings. *arXiv preprint arXiv:2008.12813*.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2023. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562*.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C-C Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6947–6965.

Lingbing Guo, Zhongpu Bo, Zhuo Chen, Yichi Zhang, Jiaoyan Chen, Yarong Lan, Mengshu Sun, Zhiqiang Zhang, Yangyifei Luo, Qian Li, et al. 2024. Mkgl: Mastery of a three-word language. *arXiv preprint arXiv:2410.07526*.

Jiabang He, Jia Liu, Lei Wang, Xiyao Li, and Xing Xu. 2024. Mocosa: Momentum contrast for knowledge graph completion with structure-augmented pre-trained language models. In *2024 IEEE International*

Conference on Multimedia and Expo (ICME), pages 1–6. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Pengcheng Jiang, Lang Cao, Cao Xiao, Parminder Bhatia, Jimeng Sun, and Jiawei Han. 2024. Kg-fit: Knowledge graph fine-tuning upon open-world knowledge. *arXiv preprint arXiv:2405.16412*.

Youmin Ko, Hyemin Yang, Taeuk Kim, and Hyunjoon Kim. 2024. Subgraph-aware training of language models for knowledge graph completion using structure-aware contrastive learning. *arXiv preprint arXiv:2407.12703*.

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

Jinpeng Li, Hang Yu, Xiangfeng Luo, and Qian Liu. 2024b. Cosign: Contextual facts guided generation for knowledge graph completion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1669–1682.

Shujie Li, Liang Li, Ruiying Geng, Min Yang, Binhua Li, Guanghu Yuan, Wanwei He, Shao Yuan, Can Ma, Fei Huang, et al. 2024c. Unifying structured data as graph for data-to-text pre-training. *Transactions of the Association for Computational Linguistics*, 12:210–228.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

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*, pages 1329–1338.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024b. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 481–484.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv preprint arXiv:1906.01195*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Moritz Plenz and Anette Frank. 2024. Graph language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4477–4494, Bangkok, Thailand. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Chenyu Qiu, Pengjiang Qian, Chuang Wang, Jian Yao, Li Liu, Fang Wei, and Eddie Eddie. 2024. 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*, pages 15257–15269.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. 2024. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6616–6626.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Martin Schmitt, Leonardo FR Ribeiro, Philipp Dufter, Iryna Gurevych, and Hinrich Schütze. 2020. Modeling graph structure via relative position for text generation from knowledge graphs. *arXiv preprint arXiv:2006.09242*.

Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. 2024. Graph transformers: A survey. *arXiv preprint arXiv:2407.09777*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Zhaoxuan Tan, Zilong Chen, Shangbin Feng, Qingyue Zhang, Qinghua Zheng, Jundong Li, and Minnan Luo. 2023. Kracl: Contrastive learning with graph context modeling for sparse knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 2548–2559.

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*, pages 1499–1509.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748.

Huijuan Wang, Siming Dai, Weiyue Su, Hui Zhong, Zeyang Fang, Zhengjie Huang, Shikun Feng, Zeyu Chen, Yu Sun, and Dianhai Yu. 2022a. Simple and effective relation-based embedding propagation for knowledge representation learning. *arXiv preprint arXiv:2205.06456*.

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, et al. 2024. Learning to plan for retrieval-augmented large language models from knowledge graphs. *arXiv preprint arXiv:2406.14282*.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022b. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167*.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022c. Language models as knowledge embeddings. *arXiv preprint arXiv:2206.12617*.

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

Derong Xu, Ziheng Zhang, Zhenxi Lin, Xian Wu, Zhihong Zhu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024. Multi-perspective improvement of knowledge graph completion with large language models. *arXiv preprint arXiv:2403.01972*.

Bo Xue, Yi Xu, Yunchong Song, Yiming Pang, Yuyang Ren, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. 2024. Unlock the power of frozen llms in knowledge graph completion. *arXiv preprint arXiv:2408.06787*.

Guangqian Yang, Yi Liu, Lei Zhang, Licheng Zhang, Hongtao Xie, and Zhendong Mao. 2024a. Knowledge context modeling with pre-trained language models for contrastive knowledge graph completion. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8619–8630.

Rui Yang, Jiahao Zhu, Jianping Man, Li Fang, and Yi Zhou. 2024b. Enhancing text-based knowledge graph completion with zero-shot large language models: A focus on semantic enhancement. *Knowledge-Based Systems*, 300:112155.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2023. Exploring large language models for knowledge graph completion. *arXiv preprint arXiv:2308.13916*.

Weihe Zhai, Arkaitz Zubiaga, Bingquan Liu, Cheng-Jie Sun, and Yalong Zhao. 2024. Towards faithful knowledge graph explanation through deep alignment in commonsense question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18920–18930.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 233–242.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020a. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3065–3072.

Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He. 2020b. Pretrain-kge: learning knowledge representation from pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 259–266.

Qian Zhao, Hao Qian, Ziqi Liu, Gong-Duo Zhang, and Lihong Gu. 2024. Breaking the barrier: utilizing large language models for industrial recommendation systems through an inferential knowledge graph. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5086–5093.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

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

Appendix

A Related Work

Knowledge Graph Completion (KGC) has evolved over the past decade and is a key task in the field of KGs. Mainstream KGC methods roughly fall into two groups: embedding-based and text-based methods. Embedding-based methods (Bordes et al., 2013; Lin et al., 2015; Sun et al., 2019; Balažević et al., 2019) generate low-dimensional vectors for entities and relations and optimize various loss functions with the goal of $h + r \sim t$ to predict missing triplets. Although simple and effective, these methods neglect the extensive textual information in KGs and struggle to handle entities and relations not encountered during training. On the other hand, text-based methods (Yao et al., 2019; Zhang et al., 2020b; Wang et al., 2022b; Liu et al., 2022; Wang et al., 2022c; Yang et al., 2024a) utilize the textual descriptions of entities and relations as input to pre-trained language models (PLMs) and introduce contrastive learning to enhance discriminative ability. However, these methods lack the inherent structural knowledge of KGs. Consequently, some efforts (Wang et al., 2021; Chen et al., 2023; He et al., 2024; Yang et al., 2024a; Qiu et al., 2024) combine embedding- and text-based KGC methods, achieving improved performance.

Graph Transformers (GTs) are essentially a special type of GNN (Bronstein et al., 2021) and are gaining increasing attention in multiple application fields (Chen et al., 2024a). In KGC, some studies (Schlichtkrull et al., 2018; Vashishth et al., 2019; Nathani et al., 2019; Chen et al., 2020; Wang et al., 2022a; Tan et al., 2023; Galkin et al., 2023) leverage GNNs to encode structural information in KGs to train embeddings for entities and relations, while initializing them with semantic embeddings via PLMs. Recently, some efforts have explored applying GTs to KG-related tasks, e.g., graph-to-text generation (Schmitt et al., 2020; Li et al., 2024c) and relation classification (Plenz and Frank, 2024). However, they either train their models from scratch or split entities and relations into multiple tokens to construct complex positional encoding matrices. For example, GLM (Plenz and Frank, 2024) is a graph transformer that fuses textual and structural information, enabling sequence PLMs to perform graph inference while maintaining their original ability.

However, GLM restricts the relative distance of individual triplets to between 0 and 32, which limits the processing of entities or relations with longer textual information. For instance, only 12.5% of triplets in WN18RR (using the T5 tokenizer) fall within this distance range. Intuitively, the constraints of integrating textual and structural information also limit the size of processable subgraphs. In addition, the attention mechanism may exhibit bias towards entities or relations with longer texts in each triplet. In this paper, we borrow the positional encoding strategy from GLM but shift our focus towards subgraph structural information while preserving GLM's strengths. We introduce a novel relative distinction matrix to achieve differentiated yet equal treatment of entities and relations in triplets. Our work is also the first to apply GT to the link prediction task.

KGC with LLMs. LLMs are deemed highly promising in the realm of KGC and have garnered extensive attention (Ren et al., 2024; Pan et al., 2024). For instance, (Yao et al., 2023; Zhu et al., 2024; Wei et al., 2024; Li et al., 2024a; Xu et al., 2024) directly perform KGC via ICL or enhance textual information in KGs to improve text-based methods. However, these methods overlook the inherent structural information of KGs, leaving LLMs unable to perceive structural knowledge. To tackle this, (Zhang et al., 2024; Liu et al., 2024b; Yang et al., 2024b) integrate structural information with LLMs to boost KGC performance. Recently, MKGL (Guo et al., 2024) enables LLMs to proficiently grasp entities and relations of KGs through three-word language, but how to make LLMs perceive graph information and improve the link prediction task remains an open problem. Going beyond the aforementioned methods, there are a handful of recent studies (Li et al., 2024b; Xue et al., 2024; Jiang et al., 2024) on leveraging LLMs for KGC.

B Complete Experimental Settings

Datasets. We evaluate different methods with three widely used KG datasets, namely FB15k-237 (Toutanova et al., 2015), WN18RR (Dettmers et al., 2018), Wikidata5M (Vrandečić and Krötzsch, 2014), for link prediction. We detail the said datasets in Table 4. Specifically, FB15k-237 is a curated dataset extracted from the Freebase (Bollacker et al., 2008) knowledge graph, covering knowledge across various domains, including movies, sports events, awards, and tourist attractions. WN18RR is a well-known dataset built from

WordNet (Miller, 1995), designed for knowledge graph research. It extracts a selection of lexical items and semantic relationships, covering a rich array of English words and their connections, such as synonyms, antonyms, and hierarchical relationships. Wikidata5M (Vrandečić and Krötzsch, 2014) is a large-scale KG dataset that integrates Wikidata and Wikipedia pages. Each entity in the dataset corresponds to a Wikipedia page, enabling it to support link prediction task for unseen entities. It follows the Wikidata identifier system, with entities prefixed by "Q" and relations by "P." Additionally, the dataset provides a text corpus aligned with the KG structure.

Baselines. To assess the effectiveness of our methods, we follow (Plenz and Frank, 2024) by using the bidirectional encoder of T5-base as the base PLM for **iGT**. Here, P and D are bucketed and mapped to B_{PD} respectively, with sharing across layers. Meanwhile, we choose three LLMs with different sizes for GLTW: Llama-3.2-1B/3B-Instruct (Dubey et al., 2024), and Llama-2-7b-chat (Touvron et al., 2023). For differentiation, we denote GLTW with different LLMs as **GLTW**$_{1b/3b/7b}$.

Also, we compare proposed GLTW and iGT against numerous embedding-based, text-based, GNN/GT-based and LLM-based baselines, which are the most relevant methods to our work. The embedding-based baselines include TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), HAKE (Zhang et al., 2020a), and CompoundE (Ge et al., 2023). The text-bsed baselines encompass KG-BERT (Yao et al., 2019), KG-S2S (Chen et al., 2022), CSProm-KG (Chen et al., 2023), and PEMLM-F (Qiu et al., 2024). The GNN/GT-based baselines cover CompGCN (Vashishth et al., 2019), REP-OTE (Wang et al., 2022a), and KR-ACL (Tan et al., 2023) (based on GNN), as well as gGLM (Plenz and Frank, 2024) (based on GT). Note that gGLM and iGT are trained on the same sampled subgraphs. The LLM-based baselines feature GPT-3.5-Turbo with one-shot ICL (marked as GPT-3.5) (Zhu et al., 2024), KG-Llama-2-13B+Struct (marked as Llama-2-13B) (Yao et al., 2023), KICGPT (Wei et al., 2024), MPIKGC-S (Xu et al., 2024), KG-FIT (Jiang et al., 2024), and MKGL (Guo et al., 2024).

Configurations. In all experiments, unless otherwise specified, we default to setting $l = 2$ and $\overline{m} = m_{hr} = m_h = m_r = m/3 = 5$ for subgraph sampling. Meanwhile, we set $\lambda = 0.5$ and

Dataset	#Ent	#Rel	#Train	#Valid	#Test
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20466
Wikidata5M	4594485	822	20614279	5133	5163

Table 4: Statistics of the Datasets. Columns 2-6 represent the number of entities, relations, triples in the training set, triples in the validation set, and triples in the test set, respectively.

$\beta_1 = 0.5$ by default. Note that β_2 is adaptively calculated based on Eq. (11). Also, we assess performance by leveraging the Mean Reciprocal Rank (MRR) of target entities and the percentage of target entities ranked in the top k ($k = 1, 3, 10$), referred to as Hits@k.

During training, we assign distinct training schedules to different modules to fully capture the knowledge in the KG datasets. These modules include iGT Encoder, LLM, Adapter and Classification Layer. Notably, we may also train the pooling operators. When training resources are limited, we follow (Guo et al., 2024) by drawing on LoRA technology (Hu et al., 2021) to mitigate memory consumption. For ease of description, we divide the training modules of GLTW into three parts: iGT Encoder, LLM, and the remaining modules (referred to as "Other Modules"). Specifically, for FB15k-237 and WN18RR, we set the number of training epochs to 10 and the gradient accumulation steps to 4. For Wikidata5M, we set the number of training epochs to 2 and the gradient accumulation steps to 10. In all experiments, we used a linear learning rate schedule and the AdamW optimizer. For iGT Encoder, LLM, and Other Modules, we set the learning rates to 0.0001, 0.00001, and 0.001, respectively, with warm-up rates (i.e., the proportion of warm-up steps to total training steps) of 0.02, 0.04, and 0.01. Given that we used three different-sized LLMs, during training, we set the batch size per device to 16 for GLTW$_{7b}$, 32 for GLTW$_{3b}$, and 64 for GLTW$_{1b}$ over WN18RR and Wikidata5M. For FB15k-237, the batch sizes are set to 32 for GLTW$_{7b}$, 64 for GLTW$_{3b}$, and 128 for GLTW$_{1b}$. Note that for all LLMs, we fine-tuned them using LoRA technology, with parameters set as follows: $r = 32$, $dropout = 0.05$, and $target\ modules = (query, value)$. We conduct our experiments on NVIDIA A800 40G GPUs with DeepSpeed+ZeRO3 and BF16. We will make the code and data publicly available upon acceptance. To ensure reliability, we report the average for each experiment over 3 random seeds.

C The construction strategy of P and D in gGLM

In this section, we introduce the positional encoding strategy of the existing method gGLM (Plenz and Frank, 2024), as shown in Fig. 5(a)–(b). Importantly, we integrate the proposed relative distinction matrix D into gGLM and illustrate an example of encoding for D in Fig. 5(c).

D KG Language Prompt

We present the KG language prompt in Table 5. Note that the prompt in Table 5 directly stems from MKGL, as our work is orthogonal to the design of the KG language prompt.

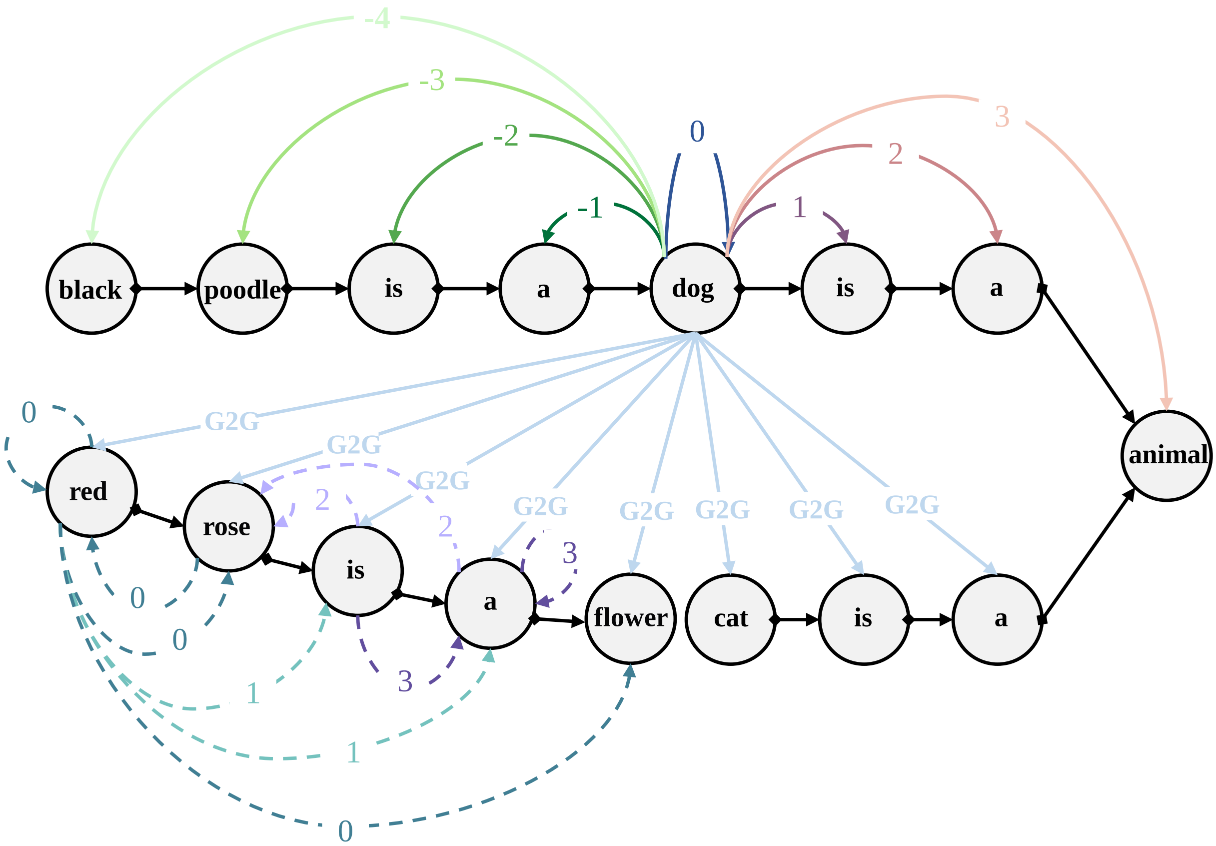
Input:
Instruction
Suppose that you are an excellent linguist studying a three-word language. Given the following dictionary:

Input	Type	Description
<*kgl:black poodle*>	Head entity	black poodle
<*kgl:is a*>	Relation	is a

Please complete the last word (?) of the sentence: <*kgl:black poodle*><*kgl:is a*>?

Response:
<*kgl:black poodle*><*kgl:is a*>

Table 5: **KG language prompt**: In the context of three-word Language, link prediction task corresponds to completing the sentence hr?. Note that we take <*kgl:black poodle*> and <*kgl:is a*> as a example.



(a) Levi graph of example subgraph with relative distances for *dog* and distinction for <red rose, is a, flower>

	black	poodle	is	a	dog	is	a	animal	cat	is	a	red	rose	is	a	flower
black	0	1	2	3	4	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
poodle	-1	0	1	2	3	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is	-2	-1	0	1	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
a	-3	-2	-1	0	1	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
dog	-4	-3	-2	-1	0	1	2	3	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is	G2G	G2G	G2G	G2G	-1	0	1	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
a	G2G	G2G	G2G	G2G	-2	-1	0	1	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
animal	G2G	G2G	G2G	G2G	-3	-2	-1	0	-3	-2	-1	G2G	G2G	G2G	G2G	G2G
cat	G2G	G2G	G2G	G2G	G2G	G2G	G2G	3	0	1	2	G2G	G2G	G2G	G2G	G2G
is	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	-1	0	1	G2G	G2G	G2G	G2G	G2G
a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	1	-2	-1	0	G2G	G2G	G2G	G2G	G2G
red	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	1	2	3	4
rose	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-1	0	1	2	3
is	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-2	-1	0	1	2
a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-3	-2	-1	0	1
flower	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	-4	-3	-2	-1	0

(b) Relative position matrix P for (a)

	black	poodle	is	a	dog	is	a	animal	cat	is	a	red	rose	is	a	flower
black	0	0	1	1	0	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
poodle	0	0	1	1	0	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is	2	2	3	3	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
a	2	2	3	3	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
dog	0	0	1	1	0	1	1	0	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
is	G2G	G2G	G2G	G2G	2	3	3	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
a	G2G	G2G	G2G	G2G	2	3	3	2	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G
animal	G2G	G2G	G2G	G2G	0	1	1	0	0	1	1	G2G	G2G	G2G	G2G	G2G
cat	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	0	1	1	G2G	G2G	G2G	G2G	G2G
is	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	2	3	3	G2G	G2G	G2G	G2G	G2G
a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	2	3	3	G2G	G2G	G2G	G2G	G2G
red	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	0	1	1	0
rose	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	0	1	1	0
is	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	2	3	3	2
a	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	2	2	3	3	2
flower	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	G2G	0	0	1	1	0

(c) Relative distinction matrix D for (a)

Figure 5: Example of subgraph preprocessing with P and D in gGLM (Plenz and Frank, 2024). Note that entries with G2G are initialized to $+\infty$.