

# Self-supervised Quantized Representation for Seamlessly Integrating Knowledge Graphs with Large Language Models

Qika Lin<sup>1</sup>, Tianzhe Zhao<sup>2</sup>, Kai He<sup>1</sup>, Zhen Peng<sup>2</sup>, Fangzhi Xu<sup>2</sup>  
Ling Huang<sup>1</sup>, Jingying Ma<sup>1</sup>, Mengling Feng<sup>1,\*</sup>

<sup>1</sup>National University of Singapore <sup>2</sup>Xi'an Jiaotong University  
qikalin@foxmail.com ephfm@nus.edu.sg

## Abstract

Due to the presence of the natural gap between Knowledge Graph (KG) structures and the natural language, the effective integration of holistic structural information of KGs with Large Language Models (LLMs) has emerged as a significant question. To this end, we propose a two-stage framework to learn and apply quantized codes for each entity, aiming for the seamless integration of KGs with LLMs. Firstly, a self-supervised quantized representation (SSQR) method is proposed to compress both KG structural and semantic knowledge into discrete codes (*i.e.*, tokens) that align the format of language sentences. We further design KG instruction-following data by viewing these learned codes as features to directly input to LLMs, thereby achieving seamless integration. The experiment results demonstrate that SSQR outperforms existing unsupervised quantized methods, producing more distinguishable codes. Further, the fine-tuned LLaMA2 and LLaMA3.1 also have superior performance on KG link prediction and triple classification tasks, utilizing only 16 tokens per entity instead of thousands in conventional prompting methods<sup>1</sup>.

## 1 Introduction

Large Language Models (LLMs), such as LLaMA (Touvron et al., 2023a,b) and GPT-4 (OpenAI, 2023), are initiating considerable transformations within the fields of artificial intelligence (AI) and natural language processing (NLP). They have achieved substantial success (Peng et al., 2023; Wang et al., 2024; Xu et al., 2024b; Luo et al., 2025), and thus, have been regarded as potential pathways towards achieving the ultimate goal of artificial general intelligence (Yang et al., 2024a). However, the specific training strategies employed

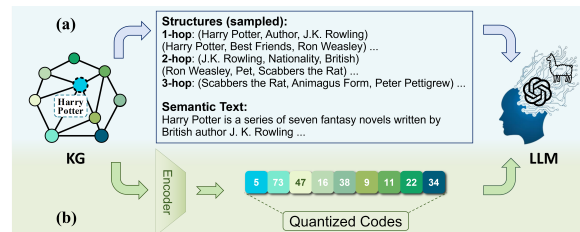


Figure 1: Illustration of different strategies to integrate KGs with LLMs. (a) The direct method utilizes (sampled) graph structures and semantic text as inputs. (b) Our method for seamlessly integrating KGs with LLMs using learned quantized and discrete codes.

by LLMs render them black-box models and struggle to retrieve the relevant facts necessary for the correct answer (Pan et al., 2024), resulting in low performance in complex reasoning scenarios (Xu et al., 2025, 2024a). Furthermore, knowledge hallucination becomes a serious issue, which may generate wrong statements that conflict with reality (Bang et al., 2023; Ji et al., 2023). It presents considerable risks, particularly in specialized fields like law (Cui et al., 2023) and healthcare (Lin et al., 2025; He et al., 2025).

Knowledge Graphs (KGs), also known as knowledge bases, organizes massive amounts of factual knowledge in a structured and interpretable manner by the triple form of (*subject, relation, object*). They can serve as a vital supplement to LLMs (Pan et al., 2024), providing an alternative way to address hallucinations and generate more precise answers using continual fine-tuning (Zhang et al., 2024b; Hron et al., 2024) or retrieve-based reasoning (Sun et al., 2024; Tan et al., 2024; Zhang et al., 2024a). However, the KGs' structure is in a graph form, which markedly differs from the discrete token format of the natural language in LLMs. Thus, due to the presence of this natural representation gap, the effective integration of comprehensive structural information of KGs with LLMs has emerged as a significant question.

As shown in Figure 1 (a), one straightforward

\*Corresponding author.

<sup>1</sup>The source codes are available at: <https://github.com/DeepReasoning/SSQR>

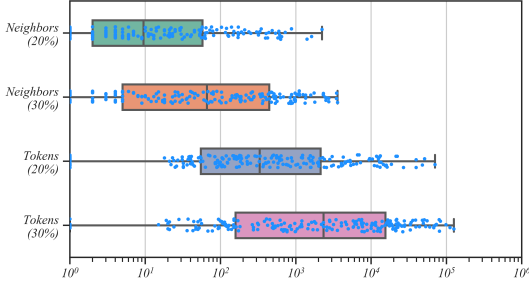


Figure 2: The statistics of 2-hop sampled neighbors and needed tokens (by LLaMA2) for entities in FB15k-237.

method involves converting relevant triples into textual prompts and then feeding them into LLMs, combined with semantic text. This simple strategy would necessitate a substantial number of tokens, causing an enormous resource burden. Supposing the average degree of an entity is  $d$ , the number of its neighbors grows exponentially and reaches  $d^h$  in the  $h$ -hop. While certain sampling strategies, such as random walk (Ko et al., 2024) and path pruning (Tan et al., 2024) have been introduced, a considerable computational load also exists. As shown in Figure 2, when only sample 20% 2-hop neighbors in FB15k-237 (Toutanova and Chen, 2015) dataset, the median and mean number of neighbors for entities are about 10 and 107, which requires median and mean tokens of about 300 and 3K, respectively. When with 30% sampling, even the median needed tokens reach about 2.5K per entity. Considering that KG tasks may involve multiple entities, even the most advanced long-context LLMs may face challenges in handling them. Meanwhile, employing KGs’ substructures through sampling could disrupt the holistic modeling of the entire graph, potentially resulting in information loss and sub-optimal performance for downstream tasks.

Another alternate strategy involves integrating continuous KG embedding with LLMs by a learnable adapter (Zhang et al., 2024b), introducing new networks in the framework. It requires additional precise alignment between the different latent representation spaces of KG embeddings and LLMs. Considering the above context, we aim to explore the potential to bridge the natural gap between KG structure and natural language and then integrate KGs with LLMs. Inspired by the early fusion strategy in multimodal LLMs (Team, 2024), the general idea of this study is to learn *compressed* and *discrete* entity codes (*i.e.*, tokens), rather than continuous embeddings, by quantized techniques to represent holistic structural and semantic information of

entities in KGs. They have the same discrete form of natural language, *e.g.*, the *quantized codes* in Figure 1 (b) align the format of language sentences. Thus, seamlessly integrating KGs with LLMs can be realized by directly inputting the learned codes into LLMs, merely requiring an expansion of the LLMs’ tokenizer vocabulary and eliminating the need for any other framework modifications.

Although several studies have conducted quantized representations on KGs (Galkin et al., 2022; Chen et al., 2023; Li et al., 2023), they universally employ an unsupervised approach to select anchors to represent entities, failing to the holistic structural and semantic modeling. In this study, we first introduce a self-supervised quantized representation for KGs, aiming to learn discrete codes for each entity that can reconstruct KG structures and align with semantic texts. A graph convolutional network (GCN) is used as an encoder to model neighbor structures of KGs, and vector quantization (Van Den Oord et al., 2017) is implemented for the KG quantized representation learning. Further, based on learned entity codes, we construct specific instructions for KG tasks, which can be seamlessly integrated with LLMs, presenting a new paradigm to employ LLMs in KG applications. In summary, our contributions lie in the following three folds:

- We propose a self-supervised quantized representation (SSQR) method that is capable of acquiring both KG structural and semantic knowledge. To our knowledge, this is the first study for KG quantization learning in a self-supervised manner.
- We propose the first study that utilizes the derived codes to seamlessly integrate KGs with LLMs, which is achieved by viewing codes as input features and designing KG instruction-following data. It has extensive potential applications, *e.g.*, KG link prediction and triple classification.
- From the experiment view, SSQR exhibits superior performance compared to current unsupervised quantized methods and the learned codes are more distinguishable. Besides, using only 16 codes for each entity, the fine-tuned LLaMA2 and LLaMA3.1 have superior performance on KG link prediction and triple classification tasks.

## 2 Quantized Representation for KGs

Formally, a KG can be represented as  $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ , which is the combination of entities  $\mathcal{E}$ , relations  $\mathcal{R}$ , and triples  $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ . Each triple is in the form of  $(h, r, t)$ . For each entity

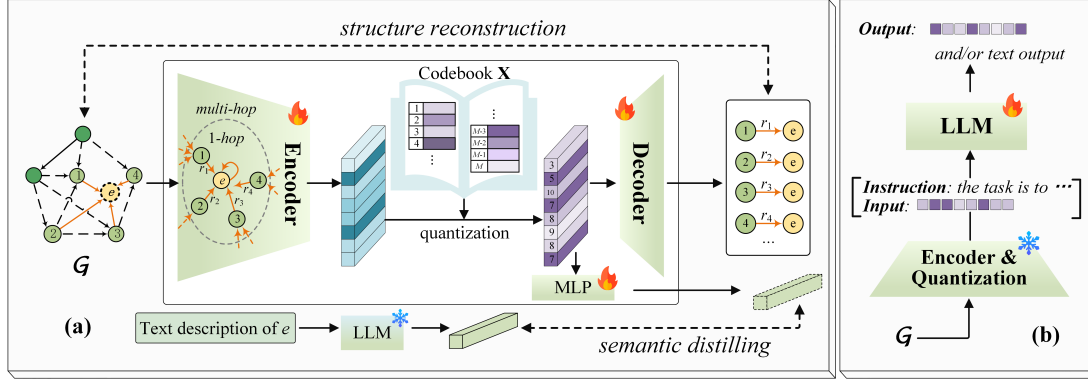


Figure 3: The overall architecture of our study. (a) is for SSQR learning. (b) is for instruction tuning for KG tasks, where the learned quantized representations serve as features. Icons \* and \* represent the status of the module during training, indicating if it is frozen or being updated, respectively.

$e$ , it has the structural and semantic information, where we utilize the entity neighbors  $\mathcal{N}(e)$  and its textual description  $T_e$  to describe, respectively. Although here we only use one-order neighbors  $\mathcal{N}(e)$  for demonstration, our model is capable of capturing high-order structure information by multi-layer GCNs. In the following contents, we will detailedly describe the structural and semantic modeling, as well as the quantized representation for KGs.

**Structural Modeling.** Here, we utilize simple but effective GCNs (Lin et al., 2022) to embed the structural information of KGs, which follows the iterative message-passing strategy to update the entity embeddings from  $l$ -th layer to  $(l+1)$ -th:

$$\mathbf{e}_j^{l+1} = \mathbf{W}_1^l \mathbf{e}_j + \sum_{(e_i, r) \in \mathcal{N}(e_j)} \mathbf{W}_2^l \mathbf{m}_{e_i, r, e_j}^l, \quad (1)$$

where  $\mathbf{e}$  is the embedding of the entity and the  $\mathbf{W}$  denotes the transformation matrix.  $\mathbf{m}$  is the message information of the specific edge. Here, we follow the composition operation (Vashishth et al., 2020) for calculation:

$$\mathbf{m}_{e_i, r, e_j}^l = \mathbf{e}_i^l * \mathbf{v}_r^l, \quad (2)$$

where  $\mathbf{v}$  is the relation embedding and  $*$  is element-wise multiplication for two vectors. Between different layers, relation representation is updated by linear transformation:  $\mathbf{v}^{l+1} = \mathbf{W}_{rel}^l \mathbf{v}^l$ . After  $L$  GCN layers, the entity representation  $\mathbf{e}^L$  and relation representation  $\mathbf{v}^L$  are all obtained.

**Quantized Representation.** Here, we introduce the quantized representation for discrete KG representation. For its implementation, inspired by VQ-VAE (Van Den Oord et al., 2017) and VQ-GAN (Esser et al., 2021), we first maintain a discrete cookbook  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$ , where

each  $\mathbf{x}_m \in \mathbb{R}^d$  is a learnable vector to represent code  $m$ . Using this, a  $d$ -dimensional vector  $\mathbf{e}$  can be quantized by matching the nearest code:

$$Q(\mathbf{e}) = \mathbf{x}_i, \text{ where } i = \arg \min_m \|\mathbf{e} - \mathbf{x}_m\|_2^2, \quad (3)$$

where  $Q$  is quantized function. In this way, each vector can be assigned to only one code, which may lack representation capacity and distinguishability for KG embedding. So we first transform the learned entity embedding  $\mathbf{e}^L$  to multiple times of dimension  $d$ , i.e.,  $\text{FFN}(\mathbf{e}^L) \in \mathbb{R}^{N \times d}$ . In this way, each entity can be assigned to a code sequence with the length of  $N$ . Thus, each entity can be represented to  $[q_1, q_2, \dots, q_N]$  by Eq. (3), where  $q_n$  is the code index in the codebook. The quantized representation vector can be:

$$\mathbf{q}_e = \mathbf{W}_q Q(\mathbf{e}^L), Q(\mathbf{e}^L) = [\mathbf{x}_{q_1}, \mathbf{x}_{q_2} \dots \mathbf{x}_{q_N}]. \quad (4)$$

Based on this, the whole model can be optimized in an end-to-end manner by the straight-through gradient estimator (Van Den Oord et al., 2017):

$$\mathcal{L}_q = \|\text{sg}[\mathbf{e}^L] - \mathbf{q}_e\|_2^2 + \beta \|\mathbf{e}^L - \text{sg}[\mathbf{q}_e]\|_2^2, \quad (5)$$

where  $\text{sg}$  stands for *stop gradient*, which is characterized by its identity function during forward computation and has zero partial derivatives for backpropagation.  $\|\text{sg}[\mathbf{e}^L] - \mathbf{q}_e\|_2^2$  is codebook loss assuring the codes are close to encoder’s outputs and  $\|\mathbf{e}^L - \text{sg}[\mathbf{q}_e]\|_2^2$  is commit loss encouraging the encoder generating outputs close to codes.  $\beta$  is a hyper-parameter to trade off the two terms.

**Structure Reconstruction.** To inject the holistic structure information into the quantized representations, we hope the learned entity codes can reconstruct KG structures. But directly predicting

adjacency matrix as is done in research for homogeneous graphs (Yang et al., 2024b) is inappropriate, because KGs’ structures are more heterogeneous and sparse. Thus, based on quantized embeddings, we verify the validity of each triplet  $(h, r, t)$  and implicitly reflect the holistic KG, where ConvE (Dettmers et al., 2018) is implemented:

$$s(h, r, t) = [\text{Flat}(\text{Conv}(\bar{\mathbf{q}}_h \| \bar{\mathbf{v}}_r^L))]^\top \mathbf{W}_c \mathbf{q}_t. \quad (6)$$

*Flat* and *Conv* are the flatten and 2D convolution operations, respectively.  $\bar{\mathbf{q}}$  and  $\bar{\mathbf{v}}$  are transformation matrices for embeddings  $\mathbf{q}$  and  $\mathbf{v}$ . The final score of triple  $(h, r, t)$  can be regularized by the sigmoid function  $\sigma$ :  $\tilde{y} = \sigma(s(h, r, t))$ . Finally, compared with actual label  $y$ , the structure modeling can be learned by binary cross-entropy loss:

$$\mathcal{L}_{st} = -\frac{1}{|\mathcal{F}|} \sum_i [y_i \log \tilde{y}_i + (1 - y_i) \log (1 - \tilde{y}_i)]. \quad (7)$$

**Semantic Distilling.** For semantic modeling, our goal is to ensure that the learned codes for each entity can imply the information of its corresponding text descriptions. Considering the substantial success of LLMs, we introduce a simple yet effective distilling strategy to learn from them. Specifically, we first obtain text embeddings of KG entities by LLMs:  $\mathbf{t}_e = \text{LLM}(T_e)$ . Here, we utilize the *text-embedding-3-large* as the LLM by OpenAI API<sup>2</sup> considering its strong ability for text embeddings. It embeds each text sequence into a 3072-dimension vector. Based on this, we make the model have the ability to align its semantic embedding through the learned quantized output, where the loss of mean square error is utilized:

$$\mathcal{L}_{se} = -\frac{1}{|\mathcal{E}|} \sum_i \|\mathbf{W}_s \mathbf{q}_{e_i} - \mathbf{t}_{e_i}\|_2^2. \quad (8)$$

In this way, we distil the semantic knowledge from the LLMs to our discrete codes of GCN outputs.

On the whole, the entire quantized representation model can be updated by the combination of quantized, structural, and semantic loss:

$$\mathcal{L} = \mathcal{L}_q + \mathcal{L}_{st} + \mathcal{L}_{se}. \quad (9)$$

### 3 Tuning LLMs with SSQR

Employing the quantized representation, each entity in KGs can be illustrated by codes of length  $N$ .

<sup>2</sup><https://platform.openai.com/docs/guides/embeddings>

**Instruction:** This is a knowledge graph completion task, which needs to predict the tail entity for an incomplete query triplet.

**Input:** The query triplet is  $(h, r, ?)$ .

The quantized representation of entity  $h$  is:  $[\text{Code}(h)]$

The answer candidates and corresponding quantized representations are as follows:

entity 1,  $[\text{Code}(\text{entity 1})]$

...

entity 20,  $[\text{Code}(\text{entity 20})]$

Please generate quantized representations of the top-3 potential answers, ranked from highest to lowest:

**Output:** 1.  $[\text{Code}(\text{candidate 1})]$

2.  $[\text{Code}(\text{candidate 2})]$

3.  $[\text{Code}(\text{candidate 3})]$

Table 1: Instruction format for link prediction, where learned codes serve as entity features to help ranking.

This can be perceived as the same form of natural language, thereby facilitating its seamless integration with LLMs. Every learned code can serve as a new token, necessitating only an expansion of the token vocabulary within the LLM’s tokenizer.

This paradigm can be applied to various KG tasks, by constructing corresponding instruction data, where the learned entity codes could act as features. For example, the KG link prediction task can be done using the instruction form as shown in Table 1. Specifically, the code sequence of entity  $e$  can be  $\text{Code}(e)$ :  $[\text{CODE}_{q_1}] [\text{CODE}_{q_2}] \cdots [\text{CODE}_{q_N}]$ . For each query  $(h, r, ?)$ , we provide the codes  $\text{Code}(h)$  for query head  $h$ . Besides, we give several answer candidates along with their associated codes for LLM ranking. Candidates can be obtained by conventional KG embedding models, e.g., TransE (Bordes et al., 2013) and CompGCN (Vashishth et al., 2020). The goal is to predict the actual ranking list of candidates using their discrete codes. The detailed instructions for triple classification are described in Section C of the Appendix. For the LLM fine-tuning, the next token prediction is carried out based on the instruction  $\mathcal{I}$  and previously generated text tokens:

$$\mathcal{L}_{llm} = -\sum_{n=1}^N \log(x_n | x_{<n}, \mathcal{I}). \quad (10)$$

### 4 Experiments and Analysis

To verify the effectiveness of the proposed SSQR and its ability to integrate with LLMs, We carry out experiments on the KG link prediction and triple classification tasks, where the popular datasets



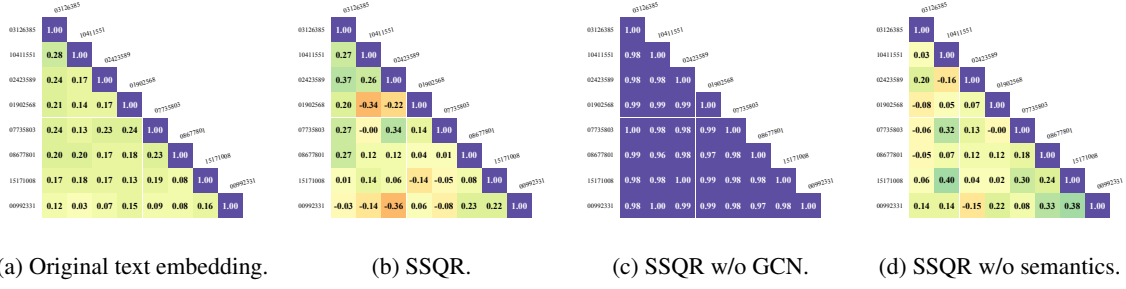


Figure 4: The cosine similarity of quantized representations on the WN18RR dataset (sampled 8 entities).

Table 2: The results of baselines are from Li et al. (2023).  $\dagger$  means the improvement of SSQR compared to the best performance in each metric.  $\ddagger$  means the ablation results compared to the results of SSQR.

Model	WN18RR		FB15k-237	
	MRR	Hits@10	MRR	Hits@10
NodePiece	0.403	0.515	0.256	0.420
+RandomEQ	0.425	0.522	0.263	0.425
EARL	0.440	0.527	0.310	0.501
+RandomEQ	0.442	0.536	0.308	0.502
SSQR	<b>0.483</b>	<b>0.578</b>	<b>0.361</b>	<b>0.545</b>
$\Delta (\dagger)^\dagger$	9.28%	7.84%	16.45%	8.57%
w/o GCN	0.479	0.577	0.309	0.482
$\Delta (\ddagger)^\ddagger$	0.83%	0.17%	14.40%	11.56%
w/o sem	0.447	0.521	0.347	0.528
$\Delta (\ddagger)^\ddagger$	7.45%	9.86%	3.88%	3.12%

WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015) as well as FB15k-237N (Lv et al., 2022) are utilized. For SSQR, a 2-layer GCN is utilized as the encoder.  $\beta$  is set to 0.25 in the experiment. The embedding dimension is set to 200 as default. The number of codebook  $M$  and codes for each entity  $N$  is set to 2048 and 32. The maximum number of training epochs is 800. For LLM fine-tuning, LLaMA2 (7B) and LLaMA3.1 (8B) are utilized using  $M$  as 2048 and  $N$  as 16 for computation efficiency. The query for head prediction  $(?, r, t)$  is transformed to the tail prediction by adding reverse relation of  $r$ . The mean reciprocal rank (MRR) and Hits@ $k$  values are set as evaluation metrics for model performance. Moreover, the triple classification task employs accuracy, precision, recall and F1-score as metrics. More detailed settings are shown in the Appendix.

#### 4.1 SSQR Results

We compare the performance of our SSQR with three unsupervised methods, *i.e.*, NodePiece (Galkin et al., 2022), EARL (Chen et al., 2023), and random entity quantization (RandomEQ

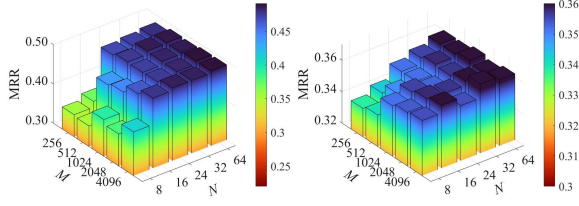
for short) (Li et al., 2023), for KG quantized representations. The results are given in Table 2.

As can be observed, SSQR achieves significant performance improvement against baselines, which has 9.28% and 7.84% improvements compared with the previous optimal performance on the WN18RR dataset. When at the FB15k-237 dataset, the improvements are even better, *i.e.*, 16.45% and 8.57%. Although these unsupervised methods are simple and efficient for implementation, they fail to capture the structures of KGs. In contrast, our proposed self-supervised strategies would provide an effective way for quantized representations for KG structure learning.

#### 4.2 SSQR Result Analysis

**Ablation Studies.** We carry out the ablation studies to verify the effectiveness of each module in SSQR as the bottom part of Table 2. Generally, the performance of link prediction degrades when GCN or semantic distilling is removed, but the extent of degradation varies across different datasets. It can be seen that the GCN encoder is more important for the FB15k-237 dataset (14.40% and 11.56% decline), while semantic information has more impact on WN18RR (7.45% and 9.86%). This may be due to the fact that FB15k-237 contains a rich KG structure which requires GCN to capture, while the semantic text is more important for WN18RR to make up for the defects caused by the lack of rich structural information.

**Relevance among Entity Codes.** We also calculate the cosine similarity of quantized representation in Figure 4, including the original text embedding, SSQR, SSQR w/o GCN, and SSQR w/o semantics. When using only text embeddings, the similarities are all small positive values. SSQR w/o GCN has similarities that are all close to 1. These phenomena indicate that entity representations are in a small corner of the space (*i.e.*, *anisotropic*), where the representation space is not fully utilized



(a) WN18RR dataset. (b) FB15k-237 dataset.

Figure 5: The effects of codebook length ( $M$ ) and sequence length ( $N$ ) for each entity.

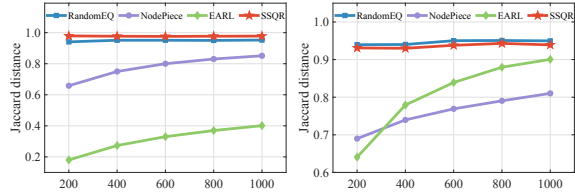
for efficient representation. SSQR solves this problem to a certain extent, with a greater range and variety of similarities. Removing semantic information would diminish that advantage.

**Impacts of  $M$  and  $N$ .** The number of codebooks and sequence lengths for codes, *i.e.*,  $M$  and  $N$ , are vital hyper-parameters for SSQR. We explore their impacts in Figure 5. Generally, larger  $M$  and  $N$  would lead to better performance as they increase the modeling ability of SSQR. In the WN18RR dataset,  $N$  has a greater impact on  $M$ , indicating the necessity of a large  $N$  for holistic and distinguishable representations. It may be caused by the sparser structure and more entities in the WN18RR.

**Distinguishability of SSQR.** Following RandomEQ, we calculate the general entropy and Jaccard distance to show codes-level and codewords-level distinguishability, respectively. For general entropy, SSQR has 16.76 and 15.27 on WN18RR and FB15k-237 datasets, similar to RandomEQ (16.75/15.27) and higher than that of NodePiece (15.94/15.26) and EARL (8.20/14.50). It shows that our method has more diverse entity codes and better entity differentiation ability than other quantization methods. The Jaccard distances of each model are shown in Figure 6. RandomEQ and SSQR have high values that are far better than those of NodePiece and EARL. RandomEQ is superior on the FB15k-237 dataset but SSQR performs better on the WN18RR dataset. In summary, SSQR exhibits a robust capacity to distinguish different entities and effectively represent KGs.

### 4.3 Quantized Representations with LLMs

**Link Prediction.** For fine-tuning, we utilize the pre-trained AdaProp (Zhang et al., 2023) to generate 20 candidates for each query as it has strong and balanced performance on most KG tasks. For comparison, we selected the current advanced embedding models, like TransE (Bordes et al., 2013), CompGCN (Vashishth et al., 2020),



(a) WN18RR dataset.

(b) FB15k-237 dataset.

Figure 6: The mean Jaccard distance between codes of a specific entity and its  $k$  nearest ones.

AdaProp (Zhang et al., 2023), MA-GNN (Xu et al., 2023), TCRA (Guo et al., 2024a), and DiffusionE (Cao et al., 2024). Besides, we include five advanced LLM-based methods for more direct comparison, including KICGPT (Wei et al., 2023), CSProm-KG-CD (Li et al., 2024), ARR (Chen et al., 2024), KG-FIT (Jiang et al., 2024), and MKGL (Guo et al., 2024b).

The results of link prediction are shown in Table 3. It can be observed that SSQR with LLaMA2 or LLaMA3.1 is obviously superior in KG link prediction against general embedding methods. Compared with the previous state-of-the-art MA-GNN, SSQR with LLaMA2 achieves about 4.60%, 8.09%, 4.39%, -0.88% and 18.47%, 32.62%, 18.31%, 4.92% improvement in two datasets, respectively.

Compared with LLM-based methods, SSQR-LLaMA2 also shows competitive performance. It is better than KICGPT, CSProm-KG-CD, and ChatGPT. Even KICGPT achieves good results on the FB15k-237 dataset, it can also be raised by 8.98%, 14.37%, 9.60%, and 7.76%. For the KG-FIT (HAKE), it also has 6.87%, 12.30%, 3.87%, and -3.16% improvements on the WN18RR dataset. Although there is a slight deficiency in terms of Hits@10, improvements on other metrics are high. Meanwhile, SSQR-LLaMA3.1 is better than SSQR-LLaMA2, demonstrating that learned quantized representations can be used for a more powerful LLM to get better performance. From all the results, our methods generally achieve a greater improvement in the Hits@1 metric, which is caused by the candidate selection and ranking strategies we used in LLM fine-tuning. The candidate selection model may have limited ability, but our method has a strong ability to select the correct answer from all candidates. This demonstrates that our method has good scalability and can be further improved with more accurate candidate selection models.

Table 3: The experiment results of general embedding methods and LLM-based methods for KG link prediction.

Model	WN18RR				FB15k237			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
General Embedding Methods								
TransE (Bordes et al., 2013)	0.223	0.014	0.401	0.529	0.330	0.231	0.369	0.528
CompGCN (Vashishth et al., 2020)	0.479	0.443	0.494	0.546	0.355	0.264	0.390	0.535
AdaProp (Zhang et al., 2023)	0.562	0.499	–	0.671	<u>0.417</u>	<u>0.331</u>	–	<u>0.585</u>
MA-GNN (Xu et al., 2023)	0.565	0.507	0.592	<u>0.679</u>	0.379	0.282	<u>0.415</u>	0.569
TCRA (Guo et al., 2024a)	0.496	0.457	0.511	0.574	0.367	0.275	0.403	0.554
DiffusionE (Cao et al., 2024)	0.557	0.504	–	0.658	0.376	0.294	–	0.539
LLM-based Methods								
KICGPT (Wei et al., 2023)	0.549	0.474	0.585	0.641	0.412	<u>0.327</u>	0.448	0.554
CSProm-KG-CD (Li et al., 2024)	<u>0.559</u>	<u>0.508</u>	0.578	0.660	–	–	–	–
ARR (Chen et al., 2024)	0.521	–	0.607	–	0.398	–	0.436	–
KG-FIT (Jiang et al., 2024)	0.553	0.488	0.595	<b>0.695</b>	0.362	0.275	<u>0.485</u>	0.572
MKGL (Guo et al., 2024b)	0.552	0.500	0.577	0.656	<u>0.415</u>	0.325	0.454	<u>0.591</u>
SSQR-LLaMA2	<b>0.591</b>	<b>0.548</b>	<b>0.618</b>	<u>0.673</u>	<b>0.449</b>	<b>0.374</b>	<b>0.491</b>	<b>0.597</b>
SSQR-LLaMA3.1	<b>0.598</b>	<b>0.559</b>	<b>0.618</b>	<u>0.675</u>	<b>0.459</b>	<b>0.393</b>	<b>0.491</b>	<b>0.597</b>

Table 4: The experiment results of the triple classification on FB15k-237N dataset. The results of baselines are taken from Zhang et al. (2024b).

Model	Acc	P	R	F1
TransE (Bordes et al., 2013)	0.697	0.708	0.671	0.689
DistMult (Yang et al., 2015)	0.587	0.590	0.568	0.579
RotatE (Sun et al., 2019)	0.684	0.692	0.664	0.678
Alpaca <sub>zero-shot</sub>	0.561	0.533	0.974	0.689
GPT-3.5 <sub>zero-shot</sub>	0.602	0.866	0.240	0.376
KG-LLaMA (Yao et al., 2023)	0.748	0.674	0.962	<u>0.793</u>
KG-Alpaca (Yao et al., 2023)	0.699	0.627	0.983	0.766
KoPA (Zhang et al., 2024b)	<u>0.777</u>	0.708	0.941	<b>0.808</b>
SSQR-LLaMA2	<b>0.794</b>	0.757	0.867	<b>0.808</b>
w/o SSQR	0.754	0.699	0.891	0.783
$\Delta$	-5.13%	-7.71%	+2.85%	-3.07%
SSQR-LLaMA3.1	<b>0.798</b>	0.759	0.872	<b>0.811</b>
w/o SSQR	0.767	0.711	0.901	0.795
$\Delta$	-3.77%	-6.34%	+3.41%	-2.03%

**Triple Classification.** Beyond the link prediction task, we conduct experiments on triple classification on the FB15k-237N dataset. The results are shown in Table 4, where our method outperforms general embedding methods and other LLM-based baselines. For the advanced KoPA (Zhang et al., 2024b) model, the performance in the F1-score metric is comparable to that of SSQR. However, the accuracies of SSQR show a significant improvement, *i.e.*, 0.794/0.798 vs. 0.777, demonstrating the effectiveness of integrating SSQR with LLMs.

#### 4.4 Insights of LLM Fine-tuning

**Ablation Studies.** We carry out ablation studies to verify the effectiveness of quantized representations for LLM tuning. The results are shown

Table 5: The ablation results for the link prediction task.

Model	MRR	Hits@1	Hits@3	Hits@10
WN18RR				
SSQR-LLaMA2	0.591	0.548	0.618	0.673
w/o SSQR	0.541	0.495	0.603	0.668
$\Delta$ ( $\downarrow$ )	8.46%	9.67%	2.43%	0.74%
FB15k-237				
SSQR-LLaMA2	0.449	0.374	0.491	0.597
w/o SSQR	0.401	0.322	0.441	0.589
$\Delta$ ( $\downarrow$ )	10.69%	13.90%	10.18%	1.34%

in Table 5 and the bottom part of Table 4, where *w/o SSQR* means only utilizing the entity’s name for fine-tuning and removing learned entity codes. For the link prediction task, there is a large performance drop, especially in the MRR, Hits@1, and Hits@3 metrics. A similar pattern is also present in the triple classification task. We observe that when under the *w/o SSQR* setting, LLMs have overfitting issues, where their performance on training sets is very high but fails to generalize to valid and test sets. This demonstrates that the learned discrete codes are distinguishable and representative for different entities, thereby allowing their utilization as features to assist KG tasks in LLMs.

**Impacts of  $M$  and  $N$  for LLM Tuning.** We explore the impacts of  $M$  and  $N$  for LLM tuning, the results are shown in Figure 7. First, we present the results of *Original*, which are the original results of AdaProp. It is shown that all other results are better than those of AdaProp, showing it is effective for LLM fine-tuning with quantized representations. The settings with  $N=16$  and  $M=2048$

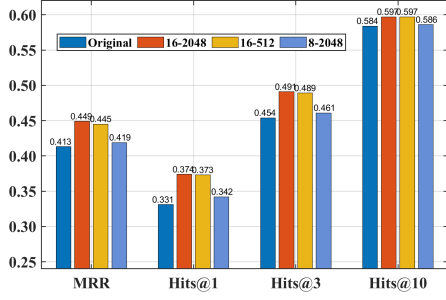


Figure 7: The impacts of quantized representation for KG link prediction task using LLMs on FB15k-237.

have better results compared to 16-512 and 8-2048, indicating large values are needed to represent entity structural and semantic information, serving better features for LLMs.  $N$  is more important than  $M$ , which drops more performance, especially on the FB15k-237 dataset (16-512 even not dropping a lot), indicating the long quantized feature is beneficial for LLMs to distinguish entities.

**Token Embeddings in LLMs.** To view the representation of codewords and actual language tokens in LLMs, we display all 2048 codewords and correspondingly sample an equal number of language tokens. Further, we reduce the embeddings to 2-dimensional space using t-SNE (Van der Maaten and Hinton, 2008) and plot the results in Figure 8. It is shown that these two types of tokens are generally divided into two categories, indicating they have different representation zones. It is consistent with the intuition and indicates LLM can perceive that they are different types of tokens, showing potential for further exploration of LLM on KG tasks using SSQR.

## 5 Related Work

For parameter-efficient embeddings on large KGs, NodePiece (Galkin et al., 2022) introduces an anchor-based method to learn a fixed-size entity vocabulary, where unsupervised strategies of Personalized PageRank (Page, 1999), node degree, and random are used for anchor selection. Each entity can be represented through  $k$  closest anchors and their respective distances. Further, EARL (Chen et al., 2023) randomly samples 10% entities as anchors and introduces connected relation information to match anchors’ counterparts. To simplify the whole process, Li et al. (2023) introduces random entity quantization (RandomEQ) to randomly set anchor entities and randomly select relations for matching. The results show that RandomEQ

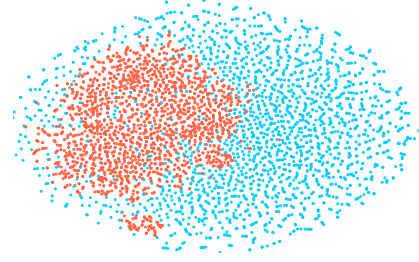


Figure 8: Token embedding virtualization in LLMs (WN18RR dataset), where red and blue dots are real word tokens and code tokens, respectively.

achieves similar results compared to previous curated strategies and has more distinguishable ability. In general, these methods are all in an unsupervised learning manner, which could be efficient for large KG embedding but fails to model comprehensive structural and semantic information.

Currently, numerous research studies are dedicated to incorporating KGs with LLMs to maximize and exploit their respective strengths (Pan et al., 2024). On one hand, using prompt engineering or retrieve strategies (Wei et al., 2023; Sun et al., 2024; Kau et al., 2024), the information of KGs be sampled and instantiated as tokens like natural language to input LLMs. On the other hand, the triple-level or sub-graph structures of KG can be inputted to the LLMs to inject knowledge (Hron et al., 2024). However, because of the natural gap between the graph structure of KGs and the natural language, how to seamlessly and effectively integrate the whole structural and semantic information of KGs with LLMs is an open problem.

## 6 Conclusion and Potential Impacts

For seamlessly integrating KGs with LLMs, we introduce a self-supervised quantized representation method (SSQR). It compresses the structural and semantic information of entities in KGs to a discrete permutation of codewords, which has a similar format as the natural language and can be directly inputted to the LLMs. By specific instruction data and fine-tuning, LLMs can seamlessly learn KG’s knowledge, which can be used in KG applications. To verify the effectiveness of our method, we implement experiments on KG link prediction and triple classification tasks, which demonstrate the superiority of our method. This innovative paradigm promises to usher in transformative techniques for KGs in the era of LLMs. In the future, we will explore more applications and make progress towards



unified frameworks for multiple KG tasks, *e.g.*, KG-based QA (Luo et al., 2024a), KG-based recommendation (Huang et al., 2023), and language modeling (Luo et al., 2024b).

## Limitations

Despite our SSQR method’s capacity to facilitate the seamless integration of KGs with LLMs, our study encounters the generalization limitation due to the substantial computational burden associated with LLMs. In most recent and our studies, LLMs are fine-tuned for a specific KG and the corresponding task, which can not be applied to various KG tasks and largely limits the model’s generalization ability. In the future, we will try to construct unified LLMs for KGs by implementing quantization within the same discrete space.

## Acknowledgments

This project was funded by the National Research Foundation Singapore under the RIE2025 Industry Alignment Fund (I2101E0002-Cisco-NUS Accelerated Digital Economy Corporate Laboratory) and under AI Singapore Programme (Award Number: AISG2-TC-2022-004).

## References

- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. [A multi-task, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity](#). In *IJCNLP*, pages 675–718.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. [Translating embeddings for modeling multi-relational data](#). In *NIPS*, pages 2787–2795.
- Zongsheng Cao, Jing Li, Zigan Wang, and Jinliang Li. 2024. [Diffusione: Reasoning on knowledge graphs via diffusion-based graph neural networks](#). In *KDD*, pages 222–230.
- Mingyang Chen, Wen Zhang, Zhen Yao, Yushan Zhu, Yang Gao, Jeff Z. Pan, and Huajun Chen. 2023. [Entity-agnostic representation learning for parameter-efficient knowledge graph embedding](#). In *AAAI*, pages 4182–4190.
- Zhongwu Chen, Long Bai, Zixuan Li, Zhen Huang, Xiaolong Jin, and Yong Dou. 2024. [A new pipeline for knowledge graph reasoning enhanced by large language models without fine-tuning](#). In *EMNLP*, pages 1366–1381.
- Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. [Chatlaw: Open-source legal large language model with integrated external knowledge bases](#). *CoRR*, abs/2306.16092.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. [Convolutional 2d knowledge graph embeddings](#). In *AAAI*, pages 1811–1818.
- Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. [Taming transformers for high-resolution image synthesis](#). In *CVPR*, pages 12873–12883.
- Mikhail Galkin, Etienne G. Denis, Jiapeng Wu, and William L. Hamilton. 2022. [Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs](#). In *ICLR*.
- Jingtao Guo, Chunxia Zhang, Lingxi Li, Xiaojun Xue, and Zhendong Niu. 2024a. [A unified joint approach with topological context learning and rule augmentation for knowledge graph completion](#). In *Findings of the ACL*, pages 13686–13696.
- Lingbing Guo, Zhongpu Bo, Zhuo Chen, Yichi Zhang, Jiaoyan Chen, Yarong Lan, Mengshu Sun, Zhiqiang Zhang, Yangyifei Luo, Qian Li, Qiang Zhang, Wen Zhang, and Huajun Chen. 2024b. [MKGL: mastery of a three-word language](#). In *NeurIPS*.
- Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. 2025. [A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics](#). *Information Fusion*, page 102963.
- Jiri Hron, Laura Culp, Gamaleldin Elsayed, Rosanne Liu, Ben Adlam, Maxwell Bileschi, Bernd Bohnet, JD Co-Reyes, Noah Fiedel, C Daniel Freeman, et al. 2024. [Training language models on the knowledge graph: Insights on hallucinations and their detectability](#). *CoRR*, abs/2408.07852.
- Qing Huang, Zhenyu Wan, Zhenchang Xing, Changjing Wang, Jieshan Chen, Xiwei Xu, and Qinghua Lu. 2023. [Let’s chat to find the apis: Connecting human, LLM and knowledge graph through AI chain](#). In *ASE*, pages 471–483. IEEE.
- 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):248:1–248:38.
- Pengcheng Jiang, Lang Cao, Cao Xiao, Parminder Bhatia, Jimeng Sun, and Jiawei Han. 2024. [KG-FIT: knowledge graph fine-tuning upon open-world knowledge](#). In *NeurIPS*.
- Amanda Kau, Xuzeng He, Aishwarya Nambissan, Aland Astudillo, Hui Yin, and Amir Aryani. 2024. [Combining knowledge graphs and large language models](#). *CoRR*, abs/2407.06564.

- Youmin Ko, Hyemin Yang, Taeuk Kim, and Hyunjoon Kim. 2024. [Subgraph-aware training of text-based methods for knowledge graph completion](#). *CoRR*, abs/2407.12703.
- Dawei Li, Zhen Tan, Tianlong Chen, and Huan Liu. 2024. [Contextualization distillation from large language model for knowledge graph completion](#). In *Findings of the EACL*, pages 458–477.
- Jiaang Li, Quan Wang, Yi Liu, Licheng Zhang, and Zhendong Mao. 2023. [Random entity quantization for parameter-efficient compositional knowledge graph representation](#). In *EMNLP*, pages 2917–2928.
- Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. [Incorporating context graph with logical reasoning for inductive relation prediction](#). In *SIGIR*, pages 893–903.
- Qika Lin, Yifan Zhu, Xin Mei, Ling Huang, Jingying Ma, Kai He, Zhen Peng, Erik Cambria, and Mengling Feng. 2025. [Has multimodal learning delivered universal intelligence in healthcare? a comprehensive survey](#). *Information Fusion*, 116:102795.
- Yang Liu, Xiaobin Tian, Zequn Sun, and Wei Hu. 2024. [Finetuning generative large language models with discrimination instructions for knowledge graph completion](#). In *ISWC*, pages 199–217.
- Haoran Luo, Haihong E, Yikai Guo, Qika Lin, Xiaobao Wu, Xinyu Mu, Wenhao Liu, Meina Song, Yifan Zhu, and Luu Anh Tuan. 2025. [Kbqa-o1: Agentic knowledge base question answering with monte carlo tree search](#). In *ICML*.
- Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, Yifan Zhu, and Anh Tuan Luu. 2024a. [Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models](#). In *Findings of the ACL*, pages 2039–2056.
- Xindi Luo, Zequn Sun, Jing Zhao, Zhe Zhao, and Wei Hu. 2024b. [Knowla: Enhancing parameter-efficient finetuning with knowledgeable adaptation](#). In *NAACL*, pages 7153–7166.
- Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. [Do pre-trained models benefit knowledge graph completion? A reliable evaluation and a reasonable approach](#). In *Findings of the ACL*, pages 3570–3581.
- OpenAI. 2023. [Gpt-4 technical report](#). *CoRR*, abs/2303.08774.
- Lawrence Page. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Technical Report.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiaapu Wang, and Xindong Wu. 2024. [Unifying large language models and knowledge graphs: A roadmap](#). *IEEE TKDE*, 36(7):3580–3599.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. [Instruction tuning with gpt-4](#). *CoRR*, abs/2304.03277.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. [Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph](#). In *ICLR*.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [Rotate: Knowledge graph embedding by relational rotation in complex space](#). In *ICLR*.
- Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2024. [Paths-over-graph: Knowledge graph empowered large language model reasoning](#). *CoRR*, abs/2410.14211.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.
- Chameleon Team. 2024. [Chameleon: Mixed-modal early-fusion foundation models](#). *CoRR*, abs/2405.09818.
- Kristina Toutanova and Danqi Chen. 2015. [Observed versus latent features for knowledge base and text inference](#). In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 57–66.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Aaron Van Den Oord, Oriol Vinyals, et al. 2017. [Neural discrete representation learning](#). *Advances in Neural Information Processing Systems*, pages 6306–6315.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-sne](#). *Journal of Machine Learning Research*, 9(11).
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. [Composition-based multi-relational graph convolutional networks](#). In *ICLR*.
- Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2024. [Boosting language models reasoning with chain-of-knowledge prompting](#). In *ACL*, pages 4958–4981.
- Yanbin Wei, Qiushi Huang, Yu Zhang, and James T. Kwok. 2023. [KICGPT: large language model with knowledge in context for knowledge graph completion](#). In *Findings of the EMNLP*, pages 8667–8683.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. [Are large language models really good logical reasoners? a comprehensive evaluation and beyond](#). *IEEE Transactions on Knowledge and Data Engineering*.

Fangzhi Xu, Qika Lin, Tianzhe Zhao, Jiawei Han, and Jun Liu. 2024a. [Pathreasoner: Modeling reasoning path with equivalent extension for logical question answering](#). In *ACL*, pages 13413–13429.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. 2024b. [Symbol-llm: Towards foundational symbol-centric interface for large language models](#). In *ACL*, pages 13091–13116.

Hongcai Xu, Junpeng Bao, and Wenbo Liu. 2023. [Double-branch multi-attention based graph neural network for knowledge graph completion](#). In *ACL*, pages 15257–15271.

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 *ICLR*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024a. [Harnessing the power of llms in practice: A survey on chatgpt and beyond](#). *ACM TKDD*, 18(6):1–32.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. 2024b. [Vqgraph: Re-thinking graph representation space for bridging gnn and mlp](#). In *ICLR*.

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2023. [Exploring large language models for knowledge graph completion](#). *CoRR*, abs/2308.13916.

Heidi C. Zhang, Sina J. Semnani, Farhad Ghassemi, Jialiang Xu, Shicheng Liu, and Monica S. Lam. 2024a. [SPAGHETTI: open-domain question answering from heterogeneous data sources with retrieval and semantic parsing](#). In *Findings of the ACL*, pages 1663–1678.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024b. [Making large language models perform better in knowledge graph completion](#). In *ACM MM*, pages 233–242. ACM.

Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. 2023. [Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning](#). In *KDD*, pages 3446–3457.

## A Statistics of WN18RR Dataset

Besides the statistic analysis of FB15k-237 in Figure 2, we also conduct the statistic analysis of WN18RR, which is shown in Figure 9. Specifically, we sample 200 entities from the whole KG and there are two settings (50% neighbor sampling and 100% neighbors). In the first setting of 50%, the median and mean of neighbors are 4.0 and 10.37, while the median and mean number of needed tokens are 61.5 and 185.84, respectively. For the setting of 100%, the median and mean of neighbors are 33.5 and 79.05, while the median and mean number of needed tokens are 623.5 and 1492.74, respectively. Compared to our SSQR, which only requires 16 tokens to represent each entity, both 50% and 100% settings demand a considerably higher number of tokens.

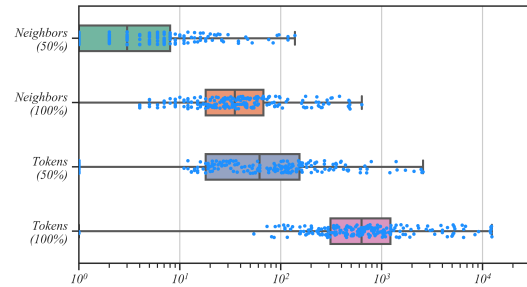


Figure 9: The statistics of 2-hop sampled neighbors and needed tokens (by LLaMA2) for entities in WN18RR.

## B Baselines

In this section, we give detailed descriptions of various baselines utilized in the paper.

### B.1 Quantized Representations for KGs

- NodePiece (Galkin et al., 2022): The selection of quantized anchors relies on unsupervised strategies, including Personalized PageRank, node degree, and random approaches.

- EARL (Chen et al., 2023): It randomly samples 10% entities as quantized anchors and introduces connected relation information to match anchors’ counterparts.

- Random entity quantization (RandomEQ for short) (Li et al., 2023): It randomly sets anchor entities and randomly selects relations for matching.

### B.2 KG Link Prediction

- TransE (Bordes et al., 2013): The strategy of incorporating translational distance is utilized for

learning representations of entities and relations.

- CompGCN (Vashishth et al., 2020): Several entity-relation composition operations are proposed to combine the semantic information of neighbor entity-relation pairs in GNNs.

- AdaProp (Zhang et al., 2023): An adaptive propagation path is learned to filter out irrelevant entities while preserving promising targets in the GNN framework.

- MA-GNN (Xu et al., 2023): A dual-branch, multi-attention-based GNN model is employed to develop expressive entity representations.

- TCRA (Guo et al., 2024a): A neuro-symbolic method that combines topological context learning with rule augmentation.

- DiffusionE (Cao et al., 2024): Introducing diffusion process to KG embedding method.

- KICGPT (Wei et al., 2023): The method utilizes a model based on embeddings as the retriever, which generates a ranked list of potential entities. An in-context learning strategy is then designed to guide ChatGPT in re-ranking these entities through multi-round interactions.

- CSProm-KG-CD (Li et al., 2024): It converts compact and structured triples into segments enriched with context by LLMs. Following this, two custom auxiliary tasks (reconstruction and contextualization) are presented, which enable compact KGC models to incorporate insights derived from these enhanced triples.

- ARR (Chen et al., 2024): A three-step (alignment, reasoning, and reranking) process designed to support and amplify conventional KG embedding models, without necessitating fine-tuning. The results are taken from the setting of LLAMA3-70B and RotatE (Sun et al., 2019) model.

- KG-FIT (Jiang et al., 2024): It involves the automatic construction of a semantically consistent entity hierarchy through clustering and LLM-guided refinement. It also details a fine-tuning technique that incorporates knowledge from the hierarchical structure and pre-trained text embeddings of entities, thereby improving KG embeddings. The results are from the HAKE model setting.

- MKGL (Guo et al., 2024b): A context retriever is introduced to help LLMs be aware of the textual and relational context of KGs. A score retriever is also used to provide the score information. LLaMA2 (7B) is utilized as the base LLM.

### B.3 KG Triple Classification

- TransE (Bordes et al., 2013): The strategy of incorporating translational distance is utilized for learning representations of entities and relations.

- DistMult (Yang et al., 2015): It utilizes the semantic matching strategy, where the validity of a fact is depicted as the matching degree between the representation of entity and relation.

- RotatE (Sun et al., 2019): It defines each relation as a rotation from the source entity to the target entity in a complex vector space.

- Alpaca<sub>zero-shot</sub>: It carries out zero-shot reasoning with Alpaca (Taori et al., 2023) with textual sequences for predicting the validity of a triple.

- GPT-3.5<sub>zero-shot</sub>: It carries out zero-shot reasoning with GPT-3.5<sup>3</sup> with textual sequences for predicting the validity of a triple.

- KG-LLaMA (Yao et al., 2023): It carries out instruction tuning with LLaMA with textual sequences for predicting the validity of a triple.

- KG-Alpaca (Yao et al., 2023): It carries out instruction tuning with Alpaca with textual sequences for predicting the validity of a triple.

- KoPA (Zhang et al., 2024b): It proposes a knowledge prefix adapter to effectively integrate pre-trained KG structural embeddings with LLMs. Alpaca-7B is utilized as the LLM backbone.

## C Experimental Details

Table 6: The statistics of WN18RR, FB15k-237, and FB15k-237N datasets. The former two are for link prediction. FB15k-237N dataset is for triple classification, where ‘/’ splits the positive and negative samples.

Dataset	Ent	Rel	Train	Valid	Test
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20466
FB15k-237N	13104	93	87282	7041/7041	8226/8226

The statistics of utilized datasets are shown in Table 6. For the SSQR learning, the default embedding dimension is set to 200. The GCN layer and dropout rate are 2 and 0.2. The training batch is 1024. For optimization, the learning rate is 0.0005 and the L2 regularization weight is 1e-8. For LLM tuning, we utilize 4 NVIDIA H100 GPUs and the learning rate is set to 2e-5 with 3% warmup ratio. In the link prediction experiment, we first tune

<sup>3</sup><https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/>



LLMs on the instruction data of CompGCN’s training split to initialize. Then, inspired by Wei et al. (2023) and Liu et al. (2024), we divide the valid set into two segments in a 9:1 ratio. The larger part is utilized to finetune LLMs to learn the ranking preference, while the smaller part is used for validation. In the triple classification experiment, we only update the embedding layer and the last four Transformer layers of LLMs for tuning efficiency. Meanwhile,  $M$  and  $N$  are set to 1024 and 16. In the training instruction data, we randomly select negative samples at a rate 16 times of positive ones. The instruction format of triple classification is shown in Table 7.

**Instruction:** Given a triple in the knowledge graph, you need to predict its validity based on the triple itself and entities’ quantized representations.  
**Input:** The triple is:  $(h, r, t)$   
The quantized representation of entity  $h$  is:  $[\text{Code}(h)]$   
The quantized representation of entity  $t$  is:  $[\text{Code}(t)]$   
Please determine the validity of the triple and respond True or False.  
**Output:** True/False

Table 7: Instruction format for triple classification.

## D Entropy and Jaccard Distance

As presented by Li et al. (2023), it is significant for the ability to distinguish different entities for quantized representations. We follow this study to calculate the entropy at the overall representation level and the Jaccard distance at the codeword-selection level. The greater entropy and Jaccard distance values denote the greater distinguishable ability. The entropy is calculated as:

$$H = - \sum p(\text{Code}(e)) \cdot \log p(\text{Code}(e)). \quad (11)$$

$p(\text{Code}(e))$  is the relative frequency of quantized representation of entity  $e$ . Moreover, the Jaccard distance is given by:

$$\mathcal{J} = \frac{1}{|\mathcal{E}| \cdot k} \sum_{e_i \in \mathcal{E}} \sum_{e_j \in k\text{NN}(e_i)} d(\text{Code}(e_i), \text{Code}(e_j)), \quad (12)$$

$$d(\text{Code}(e_i), \text{Code}(e_j)) = \frac{|\text{CSet}(e_i) \cup \text{CSet}(e_j)| - |\text{CSet}(e_i) \cap \text{CSet}(e_j)|}{|\text{CSet}(e_i) \cup \text{CSet}(e_j)|}, \quad (13)$$

where  $k\text{NN}(e_i)$  retrieves  $k$  entities. Each possesses codes that exhibit the nearest Jaccard distance to the codes of  $e_i$ .  $\text{CSet}(e)$  is the set of  $\text{Code}(e)$  by removing the order information of codewords of the entity representations.

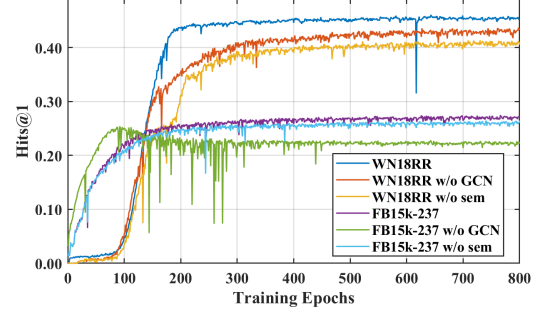


Figure 10: The training process of SSQR, where the Hits@1 metric is used to show the model performance.

## E Additional Experimental Analysis

### E.1 Training Process of SSQR

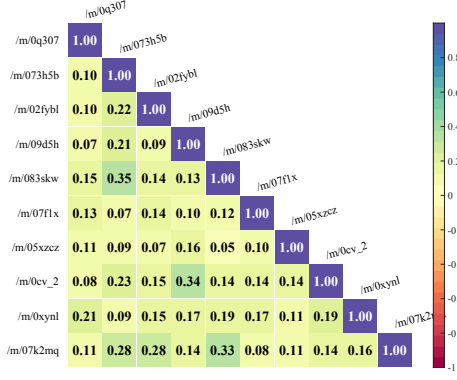
We display the training process SSQR in Figure 10, where *w/o GCN* and *w/o sem* denote ablations for the structural embedding and semantic distilling, respectively. The findings indicate that both structural embedding and semantic distilling contribute positively to the overall learning of quantized representation. The influence of semantic information on the FB15k-237 dataset is less significant when compared to its effect on GCN. Differently, semantic information is more important on the WN18RR dataset. This could be attributed to the varying levels of KGs’ sparsity.

### E.2 Relevance among Entity Codes on FB15k-237 Dataset

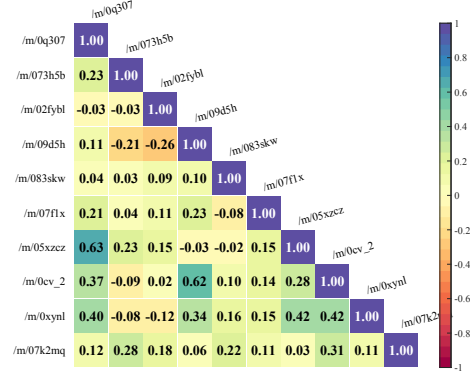
We also calculate the cosine similarity of quantized representation on the FB15k-237 dataset in Figure 11, which has the same setting as Figure 4. The contents presented in these two figures are also similar. When utilizing solely text embeddings, the corresponding similarities yield positive yet modest values. Moreover, the similarities associated with SSQR without the use of GCN are typically close to 1. These observations suggest that entity representations occupy a limited portion of the existing space, thus failing to maximize the efficiency of representation. SSQR addresses this issue to some degree by providing a broader range and diversity of similarities.

### E.3 Impacts of $M$ and $N$ for LLM Tuning on FB15k-237 Dataset

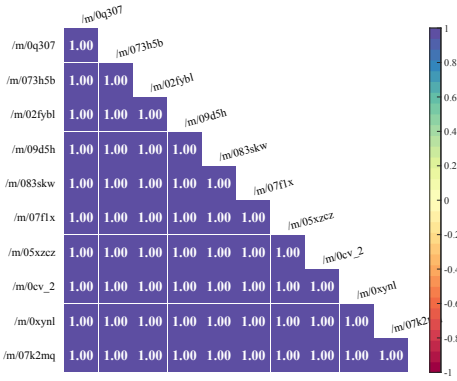
We also explore the impacts of  $M$  and  $N$  for LLM tuning on the FB15k-237 dataset, the results are shown in Figure 12. It can lead to conclusions similar to Figure 7.



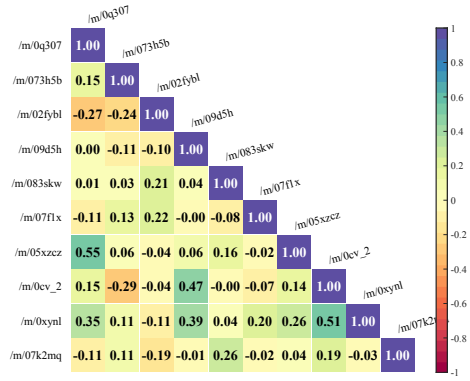
(a) Original text embedding.



(b) SSQR.



(c) SSQR w/o GCN.



(d) SSQR w/o semantics.

Figure 11: The cosine similarity of quantized representations on the FB15k-237 dataset (sampled 10 entities).

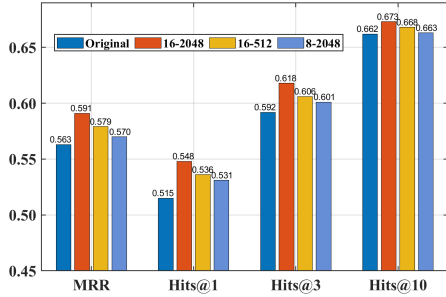


Figure 12: The impacts of quantized representation for KG link prediction task using LLMs on WN18RR.

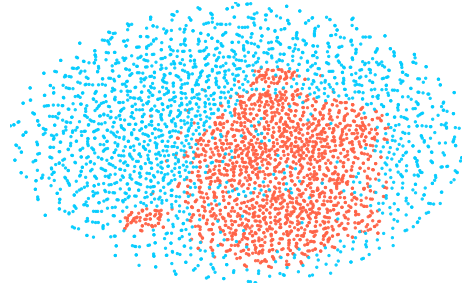


Figure 13: Token embedding virtualization in LLMs (FB15k-237 dataset), where red and blue dots are real word tokens and code tokens, respectively.

#### E.4 Token Embeddings in LLMs on FB15k-237 Dataset

Similar to Figure 8, we display the real word tokens and learned code tokens using t-SNE in Figure 13. The evidence also suggests that these two types of tokens typically fall into distinct categories, implying they each have unique representation areas.

#### E.5 Case Studies

To intuitively show the seamlessly integrating KG tasks with LLMs, we carry out case studies in Table 8, 9, and 10, covering both link prediction and triple classification tasks. It demonstrates that our method can effectively address both tasks, indicating the validity and good generalization ability of our proposed SSQR.

**Input:** This is a knowledge graph completion task, which needs to predict the tail entity for an incomplete query triplet. The query triplet is (radiotherapy, hypernym, ?).  
The quantized representation of entity radiotherapy is: [2006] [588] [350] [1486] [214] [929] [328] [1424] [1792] [919] [944] [740] [438] [843] [147] [628]  
The answer candidates and corresponding quantized representations are as follows:  
disease, [156] [1880] [1777] [185] [121] [720] [783] [1713] [945] [1077] [180] [1576] [1574] [1433] [216] [1280]  
tomography, [182] [597] [657] [1486] [404] [468] [732] [564] [833] [1470] [1756] [626] [1674] [843] [1928] [513]  
medical care, [422] [68] [1329] [1517] [1251] [431] [1479] [1445] [1666] [407] [952] [406] [1337] [388] [1982] [685]  
status, [1721] [1906] [1773] [1811] [12] [892] [1625] [1476] [1561] [176] [534] [1463] [1657] [368] [70] [1618]  
physiological state, [1721] [718] [267] [394] [120] [1105] [885] [1823] [1496] [23] [952] [406] [1559] [1198] [1149] [1800]  
medical science, [565] [413] [842] [1517] [350] [873] [575] [595] [721] [935] [1554] [175] [708] [1643] [1820] [1775]  
infection, [565] [1594] [990] [1066] [974] [40] [434] [874] [1401] [371] [1700] [1118] [1709] [52] [71] [1408]  
picturing, [788] [168] [641] [1797] [927] [711] [1608] [123] [1163] [1460] [952] [406] [1752] [1464] [553] [1158]  
medicine, [1879] [1216] [691] [296] [1743] [892] [1851] [595] [2039] [1428] [426] [740] [399] [579] [433] [1987]  
unhealthiness, [1389] [644] [570] [258] [635] [647] [732] [1139] [1660] [407] [464] [1020] [1574] [1905] [926] [1971]  
grounds, [1268] [1053] [803] [780] [1194] [285] [328] [289] [1163] [915] [1921] [1020] [524] [1774] [430] [1572]  
defense reaction, [1881] [1821] [1620] [1703] [435] [995] [908] [1308] [1596] [1598] [401] [2008] [903] [817] [92] [1158]  
radiology, [1478] [588] [1340] [1797] [1436] [1914] [1894] [1424] [634] [1460] [1756] [740] [673] [843] [108] [1088]  
radioscopy, [1005] [1002] [1441] [137] [1436] [1378] [1479] [1649] [1544] [1470] [534] [626] [902] [272] [904] [1874]  
treat, [396] [2007] [1935] [1305] [1993] [1030] [1690] [1445] [1203] [1417] [1554] [495] [1752] [1001] [1236] [98]  
specialize, [1005] [1933] [1976] [780] [927] [1728] [575] [105] [1791] [1598] [616] [1118] [1752] [425] [437] [1847]  
therapy, [396] [816] [81] [488] [336] [1164] [1690] [1288] [900] [915] [1554] [175] [666] [1622] [765] [685]  
specialism, [384] [816] [599] [394] [435] [789] [1479] [105] [664] [407] [1554] [103] [1752] [1708] [697] [1130]  
symptom, [1721] [1913] [772] [858] [120] [1150] [1374] [289] [1666] [1417] [944] [2008] [1454] [958] [1169] [1800]  
medicine, [156] [350] [1599] [1955] [1368] [508] [1527] [1445] [1561] [1460] [426] [1142] [940] [653] [793] [471]  
Please generate quantized representations of the top-3 potential answer entities, ranked from highest to lowest:  
**LLM Output:** 1, [396] [816] [81] [488] [336] [1164] [1690] [1288] [900] [915] [1554] [175] [666] [1622] [765] [685]  
2, [156] [1880] [1777] [185] [121] [720] [783] [1713] [945] [1077] [180] [1576] [1574] [1433] [216] [1280]  
3, [182] [597] [657] [1486] [404] [468] [732] [564] [833] [1470] [1756] [626] [1674] [843] [1928] [513]  
**Ground Truth:** [396] [816] [81] [488] [336] [1164] [1690] [1288] [900] [915] [1554] [175] [666] [1622] [765] [685]

Table 8: Case study on WN18RR for link prediction using LLaMA2. The code of ground truth *therapy* is ranked to the first position from 17-th.

**Input:** This is a knowledge graph completion task, which needs to predict the tail entity for an incomplete query triplet. The query triplet is (Valparaiso University, inverse relation of /location/location/contains, ?). The quantized representation of entity Valparaiso University is [527] [1345] [1849] [1227] [1751] [2038] [818] [515] [1417] [333] [29] [721] [1691] [798] [1033] [153]

The answer candidates and corresponding quantized representations are as follows:

Minnesota, [1532] [258] [1837] [357] [923] [1994] [638] [555] [771] [1003] [1736] [1473] [1495] [1436] [1313] [20]  
New York, [661] [1243] [542] [1741] [1907] [1799] [858] [1794] [1916] [458] [1844] [909] [438] [1737] [686] [963]  
California, [1059] [1286] [1604] [846] [1086] [451] [1087] [1794] [994] [297] [1463] [159] [556] [1836] [407] [963]  
Massachusetts, [202] [1243] [977] [757] [304] [389] [1172] [1308] [1916] [1858] [1323] [11] [841] [1680] [1798] [1885]  
Illinois, [961] [1025] [1267] [174] [643] [1951] [1742] [1794] [1720] [1481] [543] [1883] [695] [1921] [182] [963]  
New York City, [1458] [326] [1707] [239] [151] [640] [1366] [1794] [610] [458] [1844] [932] [122] [311] [121] [868]  
United Kingdom, [51] [193] [1354] [669] [1867] [881] [480] [1271] [392] [1858] [650] [909] [1503] [1126] [1550] [153]  
Pennsylvania, [361] [825] [1052] [1655] [1670] [732] [951] [1569] [275] [1995] [543] [4] [753] [351] [331] [637]  
Los Angeles, [1584] [1231] [1707] [1461] [1867] [1466] [265] [1933] [850] [1533] [805] [1128] [1824] [1823] [307] [963]  
Florida, [2016] [326] [542] [1614] [462] [1433] [1388] [819] [926] [1289] [1321] [563] [1977] [1144] [1268] [662]  
Ohio, [1643] [1889] [1604] [88] [1364] [485] [1819] [1569] [54] [1582] [1500] [411] [438] [125] [1636] [20]  
Texas, [2012] [1845] [1207] [412] [531] [1394] [1004] [688] [653] [1671] [1790] [1690] [1732] [1686] [1721] [1205]  
Virginia, [99] [825] [738] [1859] [1287] [1540] [708] [780] [653] [662] [756] [1873] [1514] [1686] [59] [409]  
England, [848] [1220] [1052] [590] [175] [451] [529] [1933] [808] [1598] [1790] [1988] [670] [524] [121] [136]  
Connecticut, [633] [1025] [672] [338] [1694] [1799] [1528] [1177] [1949] [458] [1703] [411] [395] [33] [233] [1013]  
Indiana, [497] [1875] [1849] [377] [1694] [61] [1471] [1445] [392] [1672] [1500] [300] [711] [1839] [331] [136]  
Los Angeles County, [662] [1772] [558] [1623] [304] [1755] [1388] [1794] [1554] [333] [1662] [1258] [1474] [1764] [1021]  
[409]  
Louisiana, [1584] [326] [1309] [50] [1035] [0] [546] [728] [1130] [297] [1284] [11] [1514] [1929] [1248] [136]  
Michigan, [172] [326] [382] [174] [1364] [1774] [1165] [664] [994] [1480] [1923] [431] [816] [1587] [1443] [860]  
Hennepin County, [584] [1890] [1263] [1168] [1990] [656] [784] [1735] [102] [1533] [1979] [1005] [1312] [786] [873]  
[1294]

Please generate quantized representations of the top-3 potential answer entities, ranked from highest to lowest:

**LLM Output:** 1, [497] [1875] [1849] [377] [1694] [61] [1471] [1445] [392] [1672] [1500] [300] [711] [1839] [331] [136]  
2, [1532] [258] [1837] [357] [923] [1994] [638] [555] [771] [1003] [1736] [1473] [1495] [1436] [1313] [20]  
3, [661] [1243] [542] [1741] [1907] [1799] [858] [1794] [1916] [458] [1844] [909] [438] [1737] [686] [963]  
**Ground Truth:** [497] [1875] [1849] [377] [1694] [61] [1471] [1445] [392] [1672] [1500] [300] [711] [1839] [331] [136]

Table 9: Case study on FB15k-237 for link prediction using LLaMA2. The code of ground truth *Indiana* is ranked to the first position from 16-th.

**Input:** Given a triple in the knowledge graph, you need to predict its validity based on the triple itself and entities' quantized representations.

The triple is: (Carl Foreman, /people/person/place\_of\_birth, Chicago)

The quantized representation of entity "Carl Foreman" is: [72] [114] [150] [336] [519] [494] [155] [482] [594] [934] [1012] [677] [473] [450] [827] [455]

The quantized representation of entity "Chicago" is: [603] [413] [385] [937] [731] [735] [836] [92] [400] [372] [563] [225] [82] [526] [81] [229]

Please determine the validity of the triple and respond True or False.

**LLM Output:** True  
**Ground Truth:** True

**Input:** Given a triple in the knowledge graph, you need to predict its validity based on the triple itself and entities' quantized representations.

The triple is: (Jessica Lange, /people/person/profession, Lawyer)

The quantized representation of entity "Jessica Lange" is: [23] [712] [484] [202] [289] [211] [772] [667] [870] [903] [213] [693] [360] [59] [868] [722]

The quantized representation of entity "Lawyer" is: [760] [84] [976] [802] [430] [735] [912] [480] [966] [411] [284] [113] [727] [744] [333] [56]

Please determine the validity of the triple and respond True or False.

**LLM Output:** False  
**Ground Truth:** False

Table 10: Two cases on FB15k-237N dataset for triple classification using LLaMA2.