# Have We Designed Generalizable Structural Knowledge Promptings? Systematic Evaluation and Rethinking

Anonymous ACL submission


## Abstract

Large language models (LLMs) have demonstrated exceptional performance in text generation within current NLP research. However, the lack of factual accuracy is still a dark cloud hanging over the LLM skyscraper. Structural knowledge prompting (SKP) is a prominent paradigm to integrate external knowledge into LLMs by incorporating structural representations, achieving state-of-the-art results in many knowledge-intensive tasks. However, existing methods often focus on specific problems, **lacking a comprehensive exploration of the generalization and capability boundaries of SKP**. This paper aims to evaluate and rethink the generalization capability of the SKP paradigm from four perspectives including **Granularity, Transferability, Scalability, and Universality**. To provide a thorough evaluation, we introduce a novel multi-granular, multi-level benchmark called SUBARU, consisting of 9 different tasks with varying levels of granularity and difficulty. Through extensive experiments, we draw key conclusions regarding the generalization of SKP, offering insights to guide the future development and extension of the SKP paradigm. Our code and data are available at this anonymous github link.


## 1 Introduction

Large language models (LLMs) (Zhao et al., 2023) have sparked a new wave in the natural language processing (NLP) field. By pre-training on massive corpus with billion-scale decoder transformers (Vaswani et al., 2017), LLMs achieve exceptional capabilities in text generation, and are widely used in current researches and applications (Zhang et al., 2024a; Chen et al., 2024; Yin et al., 2023).

However, the lack of factual accuracy in LLMs (Zhang et al., 2023) remains a significant issue, leading to unreliable and untrustworthy outputs that limit their applications. To address this, external knowledge bases are widely used (Gao et al., 2023)

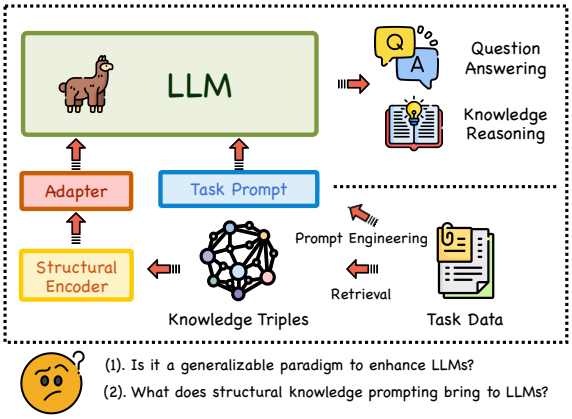


Figure 1: An intuition of the structural knowledge prompting paradigm in current LLM research.

to incorporate reliable knowledge into LLMs for a fact-grounded generation. Among these, knowledge graphs (KGs) (Liang et al., 2024) are a specialized form of semi-structured knowledge base, organizing vast amounts of factual knowledge as triples within graph structures. Numerous works (Wen et al., 2024) propose different KG-oriented methods to incorporate the KGs into the LLMs for specific tasks such as question answering (Tian et al., 2024), knowledge graph reasoning (Zhang et al., 2024b). As illustrated in Figure 1, structural knowledge prompting (SKP) is a widely used paradigm that integrates pre-trained structural information in the KGs to the LLMs with an adapter. The structural information is learned in the structural encoder, while the adapter is an extra neural network to bridge the representation gap. This approach is consistent with many Multi-modal LLMs (MLLMs) (Yin et al., 2023), where a pre-trained encoder is used to bridge the non-textual information to the textual representation space of the LLMs.

However, existing SKP methods typically adapt the paradigm directly to specific tasks with the ready-to-use principle, without thoroughly examining the paradigm itself. This raises several important questions: What makes SKP successful on

specific tasks? What level of knowledge granularity does SKP provide to LLMs, and how can it enhance generalization across tasks of varying difficulty? Furthermore, it is crucial to re-assess the full spectrum of SKP methods to better guide the future development of this research field.

To fill the gaps in current research, we explore the generalization capability of the SKP paradigm in this paper. We first construct a new **S**tr**U**ctural Prompting **B**enchm**A**rk with **R**easoning and **U**nderstanding tasks (SUBARU for short), consisting of 9 tasks with varying granularity and difficulty, which are captured from large-scale encyclopedic knowledge graphs. We design a complete training evaluation protocol to adequately assess the generalization of SKP across four dimensions: **Granularity, Transferability, Scalability, Universality**. Finally, we conduct extensive experiments with 16 different SKP settings, exploring these four dimensions and drawing key conclusions to explain the success of current SKP methods, while offering insights to guide future developments aimed at enhancing the factual accuracy of LLMs. Our contribution can be summarized as:

- We examine the widely used SKP paradigm in current LLM research. Rather than applying it to specific tasks, we provide a systematic evaluation and explore its generalization potential.
- We introduce a new benchmark, SUBARU, consisting of 9 tasks with varying granularity and difficulty levels, designed to assess the generalization of the SKP paradigm.
- We conduct a comprehensive evaluation on the generalization of existing SKP modules from four dimensions, exploring the granularity, transferability, scalability, and universality. We make several interesting conclusions after the explorations.

## 2 Related Works

The combination of KGs and LLMs (Zhang et al., 2024b; Guo et al., 2024a,b; Gutiérrez et al., 2024; Lyu et al., 2024) is an important topic in nowadays research. In addition to the factual knowledge contained in KGs, many methods try to augment the LLMs with the rich structural information present in the KG to achieve knowledge infusion capabilities. DrugChat (Liang et al., 2023) and GNP (Tian et al., 2024) employ a graph neural network to extract structural information from the retrieved knowledge subgraph to enhance the question-answering (QA) ability of LLMs. KoPA (Zhang et al., 2024b) incorporates the pre-trained structural knowledge embeddings into LLMs with a project layer to enhance the knowledge graph completion (KGC) ability of LLMs. Their paradigm lies in the use of various structural encoders to extract non-textual features and for enhancing the textual inference capability of LLM, a concept borrowed from multi-modal LLMs. While adaptations are made for specific tasks, there is a lack of in-depth exploration on the rationale of this paradigm. In this paper, we provide a comprehensive evaluation and analysis of its generalization ability.

## 3 Preliminary

In this paper, we focus on the evaluation of the structural knowledge promptings (SKP) in the LLMs. The LLM is denoted as $\mathcal{M}$. and the general prediction process of the LLM can be denoted as:

$$\mathcal{A}^* = \max_{\mathcal{A}} P_{\mathcal{M}}(\mathcal{A}|\mathcal{Q}) \tag{1}$$

where $\mathcal{A}$ is the answer to the question $\mathcal{Q}$, and $\mathcal{A}^*$ is the optimal answer decoded by the LLMs.

Many recent works aim to enhance the reasoning ability of LLMs by incorporating structural knowledge. A common approach involves retrieving relevant entities and relations from an external KG, extracting and embedding them as prompt tokens for the LLM. We denote an external KG as $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the entity set, relation set and triple set respectively. A triple $(h, r, t)$ means that there is a relation $r$ between head entity $h$ and tail entity $t$. An entity or a relation would be treated as one basic element $s_i$ in such a SKP process. For a given element $e_i$, the input prompt token can be denoted as:

$$\mathcal{S}(e_i) = \mathcal{P}(\texttt{ENC}(e_i|\mathcal{KG})) \tag{2}$$

where $\texttt{ENC}(e_i|\mathcal{KG})$ is the structural encoder learned self-supervisedly on the given $\mathcal{KG}$ and $\mathcal{P}$ is the adapter for bridging two representation spaces of the structural embeddings and LLMs. Several classic implementations of the adapter $\mathcal{P}$ exist, such as a simple MLP (Tian et al., 2024), Qformer (Li et al., 2023), etc. Depending on the specific task, the SKP tokens would be organized as a sequence $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$, which can represent a single

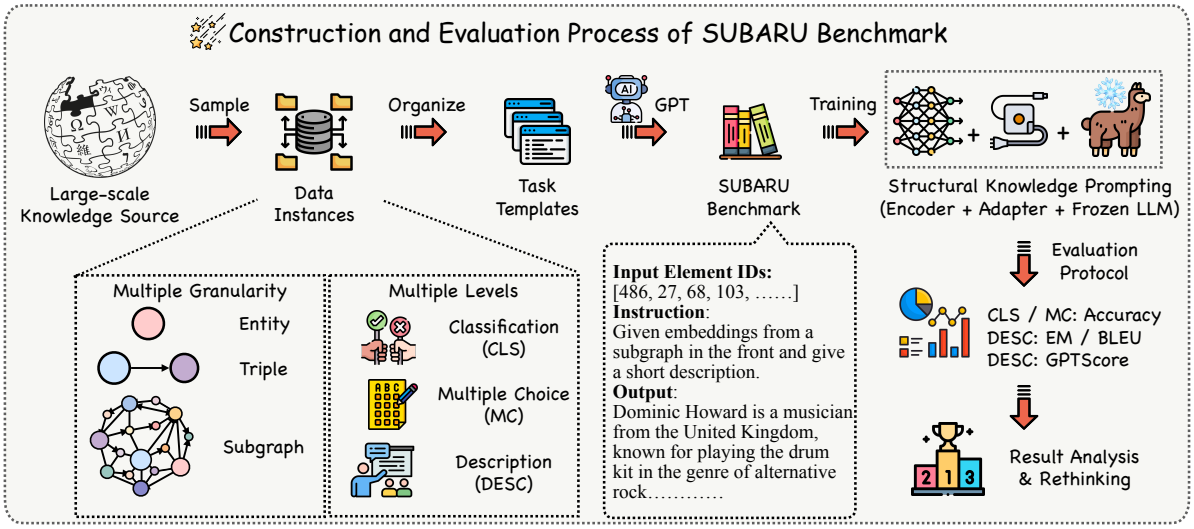


Figure 2: An overview of the construction pipeline of SUBARU benchmark and the evaluation process. In SUBARU, we construct 9 different tasks with multiple granularity (entity/triple/subgraph) and multiple difficulty levels for comprehensive evaluation of the generalization capability of the structural knowledge prompting paradigm.

entity, a triple, or a subgraph. It is further concatenated with the input tokens of the LLMs, resulting in the final prediction process:

$$\mathcal{A}^* = \max_{\mathcal{A}} P_{\mathcal{M}}(\mathcal{A}|\mathcal{Q}, \mathcal{S}) \quad (3)$$

This basic setting and formulation for SKP are widely used in existing works. However, current approaches often this paradigm directly to specific downstream tasks **without a deeper exploration of the paradigms themselves**. In this paper, we will dive deep into this problem by conducting a comprehensive evaluation with our proposed benchmark to address this gap.

## 4 Our Evaluation Framework

### 4.1 The General Motivation

In the previous section, we introduce the basic paradigm of SKP, which is widely used by current KG-enhanced LLM applications. While these methods have achieved state-of-the-art results in knowledge-intensive tasks such as QA and KGC, the generalization capabilities of the SKP paradigm remain under-explored. In this paper, we will explore the following four research questions (RQ) about the generalization ability of SKP:

- **RQ1. Granularity**: What levels of structural knowledge from KGs can the SKP paradigm integrate to LLMs?
- **RQ2. Transferability**: Is the SKP paradigm transferable across different tasks? Can SKP process new elements haven't seen before?

Table 1: The statistical information of SUBARU. We have 3 granularity and 3 levels resulting in 9 tasks.

| Task | | # Train | # Valid | # Test |
|---|---|---|---|---|
| **Entity (EG)** | CLS | 32122 | 4016 | 4016 |
| | MC | 16096 | 2012 | 2013 |
| | DESC | 16061 | 2008 | 2008 |
| **Triple (TG)** | CLS | 371168 | 20620 | 20622 |
| | MC | 185584 | 10310 | 10311 |
| | DESC | 185584 | 10310 | 10311 |
| **Subgraph (SG)** | CLS | 29454 | 3998 | 5142 |
| | MC | 14727 | 1999 | 2571 |
| | DESC | 7453 | 931 | 939 |

- **RQ3. Scalability**: Does the SKP paradigm exhibit scaling laws?
- **RQ4. Universality**: Can the SKP paradigm be applied to different LLMs?

Existing SKP methods lack the exploration of the above four questions, and no suitable benchmarks exist for such exploration. To facilitate a more thorough investigation, we start by introducing a new benchmark containing various new datasets and tasks to facilitate better nature exploration in the experimental section.

### 4.2 The SUBARU Benchmark

To better explore the mentioned four key RQs about the SKP paradigm. We propose the **S**tr**U**ctural Prompting **B**ench**m**A**r**k with **R**easoning and **U**nderstanding tasks (SUBARU for short). In this section, we we briefly introduce SUBARU, outlining its general principles and the process of

its construction. An overview of the SUBARU framework is presented in Figure 2.

#### 4.2.1 General Principle of SUBARU

Existing SKP applications typically target the specific tasks requiring varying granularity of structural knowledge from KGs. For instance, the QA task needs subgraph-level knowledge, while the KGC task only requires triple-level knowledge. Although these applications have made significant progress using the same paradigm, they **do not fully capture the capabilities of SKP**. In the SUBARU framework, we aim to evaluate the SKP paradigm more comprehensively by designing tasks with different levels of structural knowledge granularity. These include entity-granularity (EG), triple-granularity (TG), and subgraph-granularity tasks (SG), which assess the model's ability to reason and understand entities, triples, and subgraphs from KGs.

Additionally, depending on the difficulty of reasoning and comprehension, we introduce three more difficulty levels in our SUBARU: binary classification (CLS), multiple choice (MC), and description (DESC). These tasks correspond to the model's ability to perform binary classification, answer multiple-choice questions, or generate detailed descriptions based on the input structural prompts. By combining the three granularity with three levels of difficulty, we create **9 different tasks** as shown in Figure 2. Next, we describe how our dataset was constructed.

#### 4.2.2 Construction Process of SUBARU

We present an overview of the SUBARU in Table 1. We employ CoDeX (Safavi and Koutra, 2020), a large-scale KG extract from WikiData (Vrandecic and Krötzsch, 2014) as our data source. CoDeX contains approximately 110K triples. The construction process involves two key steps:

**Instance Sampling.** First, we sample entity/triple/-subgraph instances at different granularity from the KG to prepare for different tasks. For the EG task, we sample approximately 20K entities with adequate descriptions with an 8:1:1 split. For the TG task, we employ the split of CoDeX-M triples to build the datasets. For the SG task, we start with the entities selected in the EG task and then randomly sample their 1-hop and 2-hop neighborhoods to construct the subgraphs. Meanwhile, each task has specific settings. For the CLS task, we treat an entity ID with its real short name as a positive instance. For TG and SG, we consider each triple and subgraph sampled from the existing KG as positive instances. We further generate negative samples maintaining a 1:1 ratio by random perturbing. In the MC task, we sample four choices for each instance: for EG, we predict the entity name, and for TG and SG, we predict the missing entity. The missing entity prediction in TG-MC is similar to the traditional KGC task to predict the missing tail entity in the given query $(h, r, ?)$. For SG, the query provides a subgraph with one core entity missing and ask for the missing entity in the subgraph. For the DESC task, the entity, triple, and subgraph descriptions serve as the target for generation. Entity and triple descriptions are taken directly from the CoDeX dataset, while subgraph descriptions are generated using *GPT-3.5-turbo*. Due to the page limit, we present a more detailed description of the 9 tasks in Appendix A.1.

**Task Prompt Template in SUBARU**

**Input:** <Structural Knowledge Promptings $\mathcal{S}$>, <Task-specific Prompt $Q_{task}$>
**Output:** Task-specific Answers $\mathcal{A}$

Figure 3: A general prompt template for all tasks.

**Prompt Generation.** After sampling from the CoDeX KG, we create task-specific instances by applying a hand-crafted instruction prompt, $\mathcal{I}_{task}$, for each task, transforming the instances into the text format for further evaluation. Following the existing paradigms, we put the SKP in the front of the input sequence to inform the LLMs with structural information from KGs. To objectively assess the model's ability to utilize these SKPs, we remove the important textual information of the relevant elements in the instruction template, **allowing the model to complete the tasks using mainly the SKPs rather than the texts to assess the utilization of the SKPs.**. We present a general prompt template used in our evaluation in Figure 3. Here we present a general prompt template during our evaluation. We present the detailed prompt templates and data samples in Appendix A.2.

#### 4.2.3 The Evaluation Process of SUBARU

In the following experiments section, we provide a comprehensive evaluation of the four generalization properties of the SKP paradigm using the SUBARU. As SKP is an external module added to knowledge-intensive task adaption, it must be

Table 2: The main experiment results on the 9 tasks of SUBARU benchmark. We colored the top-3 results under each task with a different green color from shallow to deep.

| Method | | Entity Granularity | | | Triple Granularity | | | Subgraph Granularity | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | L-1 (Acc) | L-2 (Acc) | L-3 (EM) | L-1 (Acc) | L-2 (Acc) | L-3 (EM) | L-1 (Acc) | L-2 (Acc) | L-3 (B-4) | L-3 (GPT) |
| Random Choice | | 50.00 | 25.00 | - | 50.00 | 25.00 | - | 50.00 | 25.00 | - | - |
| FC | TransE | 55.85 | 39.49 | 0.00 | 55.41 | 87.42 | 2.77 | 82.43 | 57.95 | 9.85 | 14.97 |
| | DistMult | 52.61 | 34.02 | 0.00 | 47.23 | 89.60 | 21.21 | 87.10 | 78.99 | 4.32 | 35.42 |
| | RotatE | 57.34 | 51.16 | 0.00 | 55.59 | 66.77 | 6.88 | 70.12 | 62.54 | 2.12 | 22.20 |
| | R-GCN | 52.44 | 41.97 | 0.00 | 52.53 | 90.50 | 27.90 | 86.75 | 54.49 | 5.64 | 19.57 |
| MLP | TransE | 84.76 | 91.01 | 0.00 | 53.92 | 86.51 | 41.83 | 91.71 | 89.34 | 12.67 | 45.61 |
| | DistMult | 57.71 | 61.84 | 0.00 | 55.64 | 93.53 | 97.91 | 65.46 | 90.35 | 26.72 | 55.56 |
| | RotatE | 85.43 | 54.94 | 0.00 | 53.71 | 88.21 | 83.74 | 89.14 | 90.43 | 24.74 | 55.80 |
| | R-GCN | 66.85 | 44.46 | 0.00 | 53.68 | 90.45 | 94.59 | 76.70 | 91.05 | 16.08 | 49.25 |
| MoE | TransE | 58.66 | 38.89 | 0.00 | 65.24 | 92.07 | 19.72 | 88.46 | 80.32 | 19.50 | 46.93 |
| | DistMult | 55.35 | 27.37 | 0.00 | 54.49 | 92.92 | 8.38 | 86.98 | 81.60 | 9.75 | 35.57 |
| | RotatE | 56.47 | 29.30 | 0.00 | 59.47 | 89.09 | 7.32 | 90.31 | 88.44 | 19.68 | 47.05 |
| | R-GCN | 54.23 | 38.15 | 0.00 | 53.59 | 92.09 | 27.46 | 54.90 | 78.91 | 4.04 | 20.39 |
| Q-former | TransE | 59.48 | 92.50 | 0.00 | 54.77 | 94.42 | 38.90 | 78.97 | 27.14 | 11.32 | 41.19 |
| | DistMult | 75.34 | 60.40 | 0.00 | 52.95 | 94.11 | 17.77 | 78.59 | 37.53 | 9.25 | 32.74 |
| | RotatE | 82.96 | 79.50 | 0.00 | 50.84 | 94.02 | 6.23 | 80.35 | 26.33 | 14.43 | 42.41 |
| | R-GCN | 81.77 | 41.11 | 0.00 | 51.18 | 93.94 | 16.23 | 73.60 | 27.42 | 4.73 | 21.82 |

trained on the training set before its performance can be evaluated on the test set. The training process, based on classic next-word prediction, is defined as:

$$\mathcal{L}_{SKP} = -\log P_{\mathcal{M}}(\mathcal{A}|\mathcal{Q}_{task}, \mathcal{S}) \quad (4)$$

where $\mathcal{Q}, \mathcal{A}$ is an question-answering pair from the training data. $\mathcal{S}$ is the corresponding structural prompt for the given question, which can be an entity, a triple, or a sequential subgraph. During training, the LLM $\mathcal{M}$ is frozen, and the adapter $\mathcal{P}$ is trained to bridge the pre-trained structural encoder $\texttt{ENC}()$ and the LLM. Meanwhile, for the four RQs, we will present the detailed evaluation protocols and implementation in the next section.

## 5 Evaluation Results

In this section, we first introduce the experimental setup, including implementation details and the evaluation protocol. Then, we present the results of the experiments to explore the four significant RQs (mentioned in Section 4.1) about the Granularity, Transferability, Scalability, and Universality of the SKP paradigm. We further provide some intuitive cases to analysis the competency boundary of SKP.

### 5.1 Experimental Setup

**Implementation Details.** TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), and R-GCN (Schlichtkrull et al., 2018). We implement the structural encoders based on NeuralKG (Zhang et al., 2022). Among these structural encoders, R-GCN is a graph neural network method and others are classic KG embedding methods. For the adapter $\mathcal{P}$, we choose 4 mainstream architectures used by recent works, including single **fully-connected layer (FC)** (Zhang et al., 2024b), **multi-layer perceptron (MLP)** (Liang et al., 2023), **MoE** (Ma et al., 2024), and **Qformer** (Li et al., 2023). For LLMs, we mainly employ Llama3-8B-Instruct (Dubey et al., 2024) as the backbone model for experiments. We also evaluate the performance of other LLMs (Touvron et al., 2023) on the SUBARU in further exploration. All the experiments are conducted on a Linux server with NVIDIA A800 GPUs. We set the LLMs in FP16 precision and optimized the SKP with AdamW (Loshchilov and Hutter, 2019) optimizer. For detailed backbone selection, hyper-parameter settings, and training efficiency of all the tasks, we present a summary in Appendix B.1.

**Evaluation Protocol.** For the CLS and MC tasks, we use the accuracy (ACC) for evaluation. For the description tasks, we use different metrics for different granularity levels. For the entity-level and triple-level tasks, we use the exact match (EM) rate as the evaluation metric, which needs the LLMs to generate the exact entity name and triple information. For the subgraph-level description, we use BLEU-4 (B-4) (Papineni et al., 2002) and GPTScore as the evaluation metrics. BLEU is a traditional metric for text generation evaluation. GPTScore follows the LLM-as-a-judge paradigm (Li et al., 2024) and employs *GPT-3.5-tubor* as the judger to score the generated descriptions against

the golden answer. A more detailed introduction of our evaluation protocol and the prompt template for GPT scoring can be found in Appendix B.2.

### 5.2 Multi-Granularity Knowledge Evaluation (RQ1)

The main evaluation results are presented in Table 2. Based on these results, we make the following observations regarding the granularity of structural knowledge learned by the models:

**Observation 1. Simple MLP is surprisingly effective.** Despite recent efforts to use complex adapters for SKP, the simple MLP architecture achieves the best performance on most tasks in SUBARU. As presented in Table 2, the MLP-based results dominate in most of the colored cells compared to other adapters. Complex SKP architectures like Qformer and MoE don't perform well on certain tasks like SG.

**Observation 2. SKP excels in coarse-grained reasoning tasks.** The three granularities in SUBARU correspond to progressively coarser reasoning tasks (MC). For EG, LLMs must precisely understand the input SKP to make the correct choice. However, for TG and SG, the MC task becomes more coarse-grained, with the LLM only needing to grasp the correlation between the input SKP and the options, as the SKP provides more semantic richness and auxiliary information. We can observe that SKP performs well in the MC tasks of TG and SG than EG. This suggests that SKP demonstrates some coarse-grained reasoning ability, but lacks sufficient fine-grained understanding for EG.

**Observation 3. SKP struggles to understand new entities accurately, failing in the EG DESC task.** As we mentioned before, the smallest elements in SKP are entities or relations. Therefore, in the EG task, all the tasks require LLMs to understand the unseen entities during training and make predictions or descriptions. We observe that SKP performs poorly in the EG task, especially in the DESC task. This suggests that SKP struggles to accurately understand new entities and **lacks inductive reasoning ability at EG**. This is because the current SKP modeling approach is still relatively lacking in extrapolation capabilities. The encoder's characterization ability is inadequate for effective extrapolation when bridged with the LLM, highlighting a gap between SKP and classical MLLM in terms of generalization.

Based on these observations, we conclude that current SKP methods are **not perfect across all granularities and have their limitations**. However, mainstream SKP applications typically focus on triple or subgraph reasoning tasks, where these methods excel. The format of these downstream tasks closely resembles the TG/SG MC tasks. Additionally, to better understand the description ability of SKP models, we further conduct a case study in Figure 5.6.

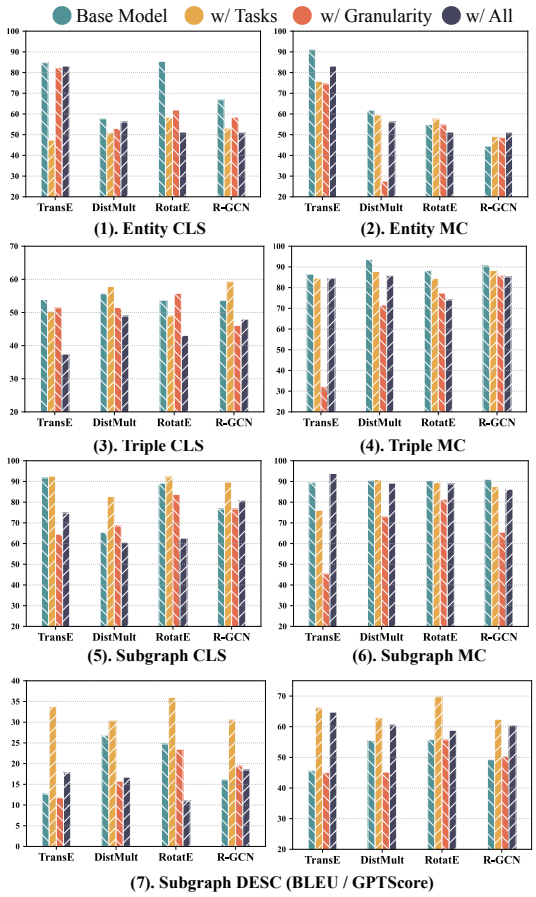


Figure 4: The transferability experiments among different granularities and levels with MLP adapter.

### 5.3 Transferability Evaluation (RQ2)

To further validate the transferability of SKP methods, we conduct an additional evaluation to answer the following two sub-issues: (1). Can SKP learn positive transfer from the tasks in different granularities and levels? (2). How well does SKP handle new entities under different scenarios? These issues relate to the **transferability of SKP across tasks and elements**.

#### 5.3.1 Transferability among Tasks

**Settings.** We conduct four sets of experiments to explore this issue by training SKP models with the dataset from the single task (Base), all tasks in the same level, same granularity, and whole benchmark

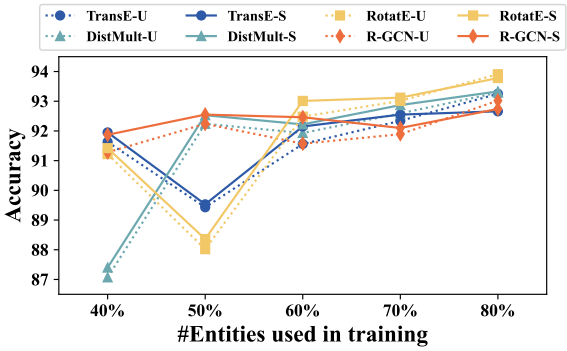


Figure 5: The inductive transfer experiments.

(w/ Tasks, w/ Granularity, w/ All). The model's performance is then evaluated on each task to investigate whether it benefits from knowledge transfer across tasks.

**Analysis.** The results are shown in Figure 4 reveal that SKP does not exhibit strong transferability on CLS and MC tasks. In most cases,training on tasks across different granularities or difficulty levels does not yield a significant improvement in model performance on the target task. However, SKP models perform better on the DESC task when trained with additional data, likely due to the nature of the task. SG DESC is a relatively coarse-grained generation task that can benefit from extra training data containing more structural information.

Overall, this experiment suggest that the current SKP architecture faces challenges in transferability.

#### 5.3.2 Transferability in New Elements

To explore SKP's ability to handle new entities, we conduct experiments in an inductive transfer scenario for TG tasks. As we mentioned before, the entity is the basic element in SKP, making EG tasks naturally inductive. While SG Task will inevitably have overlapping elements, TG is the best scenario for inductive experiments.

**Settings.** We re-split the datasets by preserving a certain ratio (%) of entities in the training set. In the test set, there will be some unseen entities. This allows us to further divide the test triples into two categories: triples with unseen entities (U) and those without (S). We evaluate the model's performance on these two subsets separately.

**Analysis.** The results in Figure 5 suggest that SKP performs well with new entities in the TG MC task. Specifically, the performance on the unseen triples is nearly identical to that on the seen triples. Besides, training on more entities can improve the model's ability to inductive reasoning. This raises an interesting question: why does SKP perform better in terms of transferability on TG tasks compared to EG tasks? We believe this is due to a combination of task difficulty and granularity. The SKPs in TG and SG tasks provide more structural context, which reduces the difficulty for the model in understanding the key elements, making it easier to generate correct predictions.

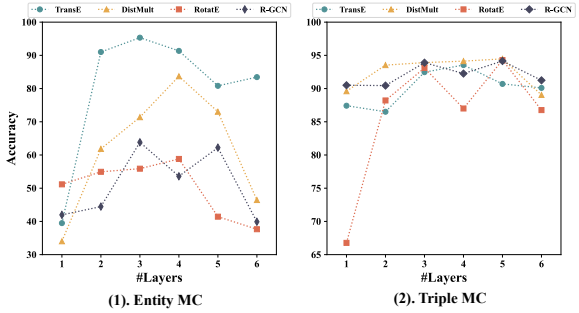


Figure 6: The scalability experiments on the entity / triple MC tasks. We use MLP as the adapter.

### 5.4 Scalability Evaluation (RQ3)

To verify the scalability, we conducted the following experiment.

**Settings.** We dedicate research to the MLP adapter, constantly deepening its layers from 1 to 6 for experiments on the SUBARU benchmark.

**Analysis.** The results presented in Figure 6 reveal that adapters in SKP models undergo an upward and then downward change in performance. An MLP with 3-4 layers is strong enough to achieve state-of-the-art performance. This law is confirmed on different structural encoders. Of course, the scalability of the adapter is also related to the amount of data required for training, which is currently limited, and the scalability at larger data volumes needs to be further explored.

### 5.5 Universality Evaluation (RQ4)

**Settings.** To further evaluate the universality of SKP, we conduct experiments with more different LLM backbones such as Llama2-7B-chat (Touvron et al., 2023), Llama3.1-8B-Instruct (Dubey et al., 2024), and Mistral-7B (Jiang et al., 2023). We report the results of them on the MLP adapter.

**Analysis.** As shown in Figure 7, we can find that SKP is a universal framework that can be adapted in various decoder LLMs. The performance variation across different backbones on the same task is minimal, and a consistent trend is observed across models. However, there is a slight performance drop when combining certain backbones, likely due to differences in the LLM architectures. More

Table 3: A case study for subgraph description task. We colored the key words of the sentence.

| |
|---|
| **(Golden Answer) Franz Grashof** was a university teacher in the field of mechanics, who spoke, wrote, or signed in German. He was buried at Hauptfriedhof Karlsruhe. |
| **(Good Prediction) Friedrich Ernst** Petersen was a German engineer who worked in the field of engineering. He was employed at the University of Göttingen, where he was also educated. Friedrich Ernst Petersen spoke, wrote, or signed in German. **(GPTScore: 90)** |
| **(Neutral Prediction) Friedrich August** von Hayek was a university teacher who specialized in the field of economics. He was fluent in German, which he spoke, wrote, and signed. After his death, he was buried in the cemetery of the village of Zermatt. **(GPTScore: 60)** |
| **(Bad Prediction) Karl-Heinz Rummeny**, a notable alpine skier, was born in Garmisch-Partenkirchen. He was also a member of the German Alpine Club and was also a member of the German Ski Association. **(GPTScore: 25)** |

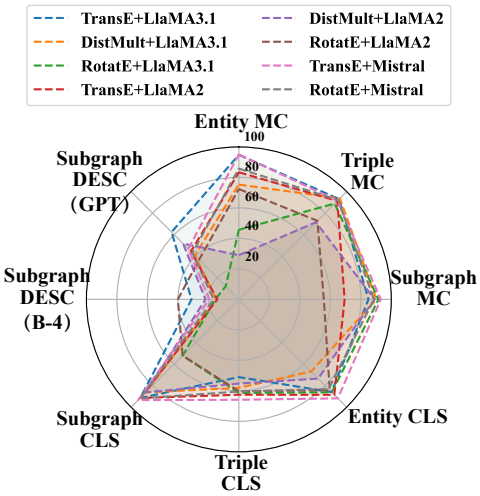


Figure 7: Experiments on more different LLMs.

additional results about LlaMA2 and LlaMA3.1 are presented in Appendix B.4.

### 5.6 Case Study and Further Analysis

In the three difficulty levels we designed in SUBARU benchmark, the CLS and MC tasks provide clear answers and quantitative metrics, allowing for precise comparisons of model performance. However, for the task of subgraph DESC, assessing the quality of generated text is more subjective. Therefore, we conduct case studies in this section to analyze the ability of the SKP model to describe the subgraph structures. **The goal of this case study is not to compare the differences in the performance of different SKP models, but to identify the commonalities that exist in descriptions.** As shown in Figure 3, we present a simple case with a golden answer and several predictions from different SKP models, which would be a personal description. We can observe the following two points:

(1). None of the SKP models were able to accurately identify the central entity, highlighting the **inability to include particularly precise and personalized information** in the SKPs. This also explains why all the SKP models fail in the EG DESC task in Table 2, which requires precise entity identification.

(2). **The SKP models demonstrate an understanding of some coarse-grained entities and relations** in the input SKP, capturing their connections and reflecting semantic understanding in the generated text. A good prediction can understand more hidden information encoded in SKP such as profession, major, nationality, and skills.

Combining these insights, we can conclude that SKP can inform LLMs coarse-grained information for understanding some subgraph structures roughly, , but struggles with detailed information like specific names, places, or specialized terms. While SKP excels at recognizing broad knowledge such as entity attributes, it lacks the cognitive ability to handle finer details. As text generation and deep-level understanding are key capabilities for LLMs, we think that **future improvements to SKP should focus on activating more precise and detailed information through additional prompt tokens**.

## 6 Conclusion

In this paper, we investigate a popular paradigm SKP which aims to integrate external structural knowledge into LLMs. We conduct a thorough evaluation of its generalization capabilities using a new benchmark, SUBARU, which encompasses multiple levels of granularity and difficulty. We detail the construction process and evaluation protocol of SUBARU. After conducting sufficient experiments in four perspectives, we draw several insightful conclusions. Our findings suggest that SKP effectively provides LLMs with coarse-grained information across different granularities and task types. However, **achieving fine-grained, precise factual awareness remains a significant challenge**. This evaluation will guide the future development of the SKP to incorporate multiple granularity structural knowledge and task-solving abilities into LLMs.

## Limitations

In this paper, we make a deep exploration of the generalization of the structural knowledge prompting paradigm. Our work has the following three limitations:

**The scale of the SUBARU benchmark.** The benchmark constructed by us has some limitations in terms of scale. SUBARU does not consist of million-scale training and evaluation data, which limits the exploration of the scalability.

**The exploration on larger LLMs.** Due to the limited computational resources, we mainly conduct experiments on LLMs with 7B-8B parameters. Though most of the SKP works are based on LLMs with the same scale, our exploration lacks results on larger LLMs such as 13B and 70B LlaMA.

**Lack of explanation of internal mechanisms.** We mainly evaluate the SKP paradigm by the tasks and metrics defined by the SUBARU benchmark, lacking further exploration of the internal mechanisms in LLMs, such as the layer-wise attention weights analysis.

We will continue to solve these limitations.

## Ethical Considerations

In this paper, all of our research and experiments are conducted on publicly available open-source datasets and models. We construct our evaluation benchmark from open-source data and we will release them for open research. Therefore, there is no ethical consideration in this paper.

## References


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.

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. 2024. On softmax direct preference optimization for recommendation. *CoRR*, abs/2406.09215.

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*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *AISTATS*, volume 15 of *JMLR Proceedings*, pages 315–323. JMLR.org.

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. 2024a. Mkgl: Mastery of a three-word language.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024b. Lightrag: Simple and fast retrieval-augmented generation. *CoRR*, abs/2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *CoRR*, abs/2405.14831.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv: 2411.16594*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, Fuchun Sun, and Kunlun He. 2024. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Youwei Liang, Ruiyi Zhang, li Zhang, and Pengtao Xie. 2023. Drugchat: Towards enabling chatgpt-like capabilities on drug molecule graphs. *TechRxiv*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net.

Yougang Lyu, Lingyong Yan, Shuaiqiang Wang, Haibo Shi, Dawei Yin, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2024. Knowtuning: Knowledge-aware fine-tuning for large language models. In *EMNLP*, pages 14535–14556. Association for Computational Linguistics.

Qiyao Ma, Xubin Ren, and Chao Huang. 2024. Xrec: Large language models for explainable recommendation. In *EMNLP (Findings)*, pages 391–402. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Tara Safavi and Danai Koutra. 2020. Codex: A comprehensive knowledge graph completion benchmark. In *EMNLP (1)*, pages 8328–8350. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR (Poster)*. OpenReview.net.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *AAAI*, pages 19080–19088. AAAI Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10370–10388. Association for Computational Linguistics.

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 (Poster)*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *CoRR*, abs/2306.13549.

Wen Zhang, Xiangnan Chen, Zhen Yao, Mingyang Chen, Yushan Zhu, Hongtao Yu, Yufeng Huang, Yajing Xu, Ningyu Zhang, Zezhong Xu, Zonggang Yuan, Feiyu Xiong, and Huajun Chen. 2022. Neuralkg: An open source library for diverse representation learning of knowledge graphs. In *SIGIR*, pages 3323–3328. ACM.

Yichi Zhang, Zhuo Chen, Yin Fang, Yanxi Lu, Fangming Li, Wen Zhang, and Huajun Chen. 2024a. Knowledgeable preference alignment for llms in domain-specific question answering. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 891–904. Association for Computational Linguistics.

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 Multimedia*, pages 233–242. ACM.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

# A Details in SUBARU benchmark

## A.1 Detailed Task Settings

In this section, we provide a detailed description of the 9 task settings in the SUBARU benchmark. Note that we have 3 different granularity (EG/TG/SG) and 3 different levels (CLS/MC/DESC) in the SUBARU benchmark.

- **Task 1: EG CLS.** This task needs the LLM to predict the true or false of a given question about whether a given embedding and an entity name are a pair.
- **Task 2: EG MC.** This task needs the LLM to select the true answer in the given 4 options to answer the question about a given entity embedding.
- **Task 3: EG DESC.** This task needs the LLM to generate a short entity name to answer what the entity is based on the input SKP.
- **Task 4: TG CLS.** This task needs the LLM to predict the true or false of a given question about whether the SKP of the triple is a positive one.
- **Task 5: TG MC.** This task needs the LLM to select the true answer in the given 4 options to complete the given query in the form of SKP. The query can be a head prediction $(?, r, t)$, a relation prediction $(h, ?, t)$, or a tail prediction $(h, r, ?)$. Here, we denote $?$ as the missing entity/relation that needs to be completed by the model.
- **Task 6: TG DESC.** This task needs the LLM to generate the head/tail entity name and the relation to answer what the triple is based on the input SKP.
- **Task 7: SG CLS.** This task needs the LLM to predict the true or false of a given question about whether the SKP of the subgraph is a positive one.
- **Task 8: SG MC.** This task needs the LLM to select the true answer in the given 4 options to complete the given query in the form of SKP. The query is a subgraph that removes a key entity, which should be predicted by the model.
- **Task 9: SG DESC.** This task needs the LLM to generate a paragraph to describe what the SKP is in the given input. The subgraph is extracted from the KGs by random sampling.

Note that, for CLS tasks, an entity-short name pair/triple/subgraph sampled from the KG is regarded as a positive sample. We generate negative samples by randomly replacing the positive samples in a 1:1 manner. For the EG DESC and TG DESC tasks, the golden label for each entity and relation is their short name in the given KG. For the SG task, we employ GPT-3.5 to generate golden answers. The prompt template we used is presented in Figure 8. We manually verified the generated results and found that the generated golden answer is of acceptable quality and can be used to train models of around 7B.

## A.2 The Prompt Templates

We present the prompt templates we used in the SUBARU benchmark in Figure 10 to Figure 18. For each task, the instruction is consistent and the input would be changed by different data instances. We present one case for each task.

# B Experiments

## B.1 Experimental Details

In our experiments, we implement the training and evaluation process with PyTorch (Paszke et al., 2019) and hugging-face transformers (Vaswani et al., 2017) library. We train 3 epochs for each SKP model with a fixed context length of 384. The batch size is set to 16. We tune the learning rate in $\{1e^{-4}, 3e^{-4}, 5e^{-4}\}$.

For the structural encoders, we set the embedding dimensions of four different backbones to 512. We implement them using NeuralKG (Zhang et al., 2022), with a 3000 epoch training until coverage. The KG embedding methods (TransE/DistMult/RotatE/RGCN) are classic backbones to train structural embedding for a given KG. Besides, R-GCN (Schlichtkrull et al., 2018) employs a relational graph convolution layer for message aggregation in the KG. The training process is self-supervised. The training objective can be denoted as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{(h,r,t)\in\mathcal{T}} \Big( -\log\sigma(\gamma - \mathcal{F}(h,r,t)) - \sum_{i=1}^{K} p_i \log\sigma(\mathcal{F}(h'_i, r'_i, t'_i) - \gamma) \Big) \quad (5)$$

**Prompt for Golden Answer Generation**

Given several triples in an extracted subgraph from a knowledge graph, you need to organize them into text paragraphs to describe the information contained in this graph.
The given triple:
<head_1, relation_1, tail_1>
<head_2, relation_2, tail_2>
.....
<head_n, relation_n, tail_n>
Your Answer:

Figure 8: The prompt template used to generate the golden answer of SG DESC task.

**GPT Evaluation Prompt Template**

Score the given model-generated text against the ground truth on a scale from 0 to 100, focusing on the alignment of meanings rather than the formatting.
**The ground truth text**: <Golden Label>
**The model output**: <Model Prediction>
Provide your score as a number and do not provide any other text in the response.

Figure 9: The prompt template used for GPT evaluation.

where $(h, r, t) \in \mathcal{T}$ is a positive triple. $\sigma$ is the sigmoid function and $\gamma$ is a margin hyper-parameter. $p_i$ is the self-adversarial training weights proposed by RotatE (Sun et al., 2019). $\mathcal{F}$ is the score function defined specifically by different methods. For example, the score function of TransE is:

$$\mathcal{F}(h, r, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_1 \tag{6}$$

For the four kinds of adapters, we implement them with the following setting:

- **FC**. It is implemented by a single linear layer in PyTorch in the form of $d_e \times d_l$, where $d_e$ is the structural embedding dimension and $d_l$ is the token embedding dimension of LLMs.

- **MLP**. It is implemented by several linear layers with ReLU (Glorot et al., 2011) as an activation function. The intermediate dimension of the MLPs is $3 \times d_e$. In most of the experiments, we use a two-layer MLP. In the scalability experiments, we explore deeper MLPs.

- **MoE**. We follow the implementation in XRec (Chen et al., 2024) for the MoE adapter layers. We set the expert number to 4 with adaptive gated fusion.

- **Qformer**. We follow the implementation in BLIP-2 (Li et al., 2023). The number of transformer layers in Qformer is set to 2 with 2 attention heads. The readout layer is set to be a two-layer MLP.

Now we can explain why we chose these four kinds of adapters in our evaluation. Note that the paradigm of SKP and MLLMs have certain ideas in common, which is **bridging heterogeneous information into LLM with adapters and employing texts as a core expression to solve different tasks**. Therefore, existing SKP models are heavily informed by MLLMs.

Overall, they are all popular architectures used by existing methods. FC and MLP are fundamental neural networks used by GNP (Tian et al., 2024) and KoPA (Zhang et al., 2024b). MoE network is used by XRec, a work that attempts to inform LLMs with the structural information in user-item interaction graphs. Though different from the structural information in KGs, it is also worth a try. Qformer is a classic adapter widely used in Multimodal LLMs such as BLIP-2, which is more complex than vanilla FC and MLP. Though no current work employs Qformer, we think Qformer is a representative design with amazing ideas. Therefore, we also evaluate it in our experiments.

Table 4: A further exploration on the influence of textual query in the TG MC task. We use the MLP as the adapter for SKP.

| Method | w/ text | w/o text |
|---|---|---|
| **TransE** | 86.51 | 49.09 |
| **DistMult** | 97.91 | 90.94 |
| **RotatE** | 83.74 | 83.51 |
| **R-GCN** | 94.59 | 78.45 |

Table 5: Results on LlaMA3.1-8B.

| Method | | Entity MC | Triple MC | Subgraph MC |
|---|---|---|---|---|
| **FC** | **TransE** | 51.86 | 88.26 | 49.35 |
| | **DistMult** | 30.15 | 90.25 | 78.10 |
| | **RotatE** | 35.67 | 72.65 | 46.28 |
| | **R-GCN** | 40.04 | 89.12 | 65.61 |
| **MLP** | **TransE** | 94.83 | 93.04 | 85.05 |
| | **DistMult** | 75.31 | 93.67 | 90.19 |
| | **RotatE** | 45.65 | 88.84 | 91.59 |
| | **R-GCN** | 67.81 | 93.78 | 90.93 |
| **MoE** | **TransE** | 44.16 | 89.94 | 78.56 |
| | **DistMult** | 37.31 | 91.50 | 76.93 |
| | **RotatE** | 61.00 | 91.60 | 81.83 |
| | **R-GCN** | 46.24 | 90.54 | 68.84 |
| **Q-former** | **TransE** | 88.47 | 90.95 | 20.34 |
| | **DistMult** | 75.21 | 93.14 | 44.18 |
| | **RotatE** | 72.67 | 92.67 | 28.54 |
| | **R-GCN** | 69.64 | 92.83 | 22.59 |

The training process task about 20 to 30 minutes for EG and SG tasks in our experiment environment while TG takes about 4 hours. Some of our code implementation is under the help of AI assitant like ChatGPT.

### B.2 Evaluation Details

In our evaluation protocol, we have several different metrics for different tasks. The CLS tasks and MC tasks have deterministic results, which can be measured by the quantitative accuracy metric. For the DESC, the situation becomes more complex. This is caused by the different settings in the DESC tasks. For EG DESC, we expect the model to generate a precise entity name. For TG DESC, we expect the model to generate the entities and relations properly in the given triple. For the SG DESC, the target is to create a paragraph to describe the given subgraph context. We can find that among these tasks, EG and TG require high accuracy and need to use Exact Match (EM) as an evaluation metric. SG, although also requires high accuracy, is not suitable for EM due to the generation of long text, so we adopt the current model of combining BLEU and GPTScore to evaluate the semantic similarity of the generated texts and the golden labels. The prompt template we used in the GPT evaluation is presented in Figure 9.

Table 6: Results on LlaMA2-7B-chat.

| Method | | Entity MC | Triple MC | Subgraph MC |
|---|---|---|---|---|
| **FC** | **TransE** | 34.72 | 46.77 | 43.29 |
| | **DistMult** | 26.08 | 61.75 | 70.75 |
| | **RotatE** | 23.00 | 67.21 | 52.66 |
| | **R-GCN** | 18.18 | 64.62 | 50.91 |
| **MLP** | **TransE** | 83.38 | 91.32 | 69.38 |
| | **DistMult** | 29.01 | 71.51 | 89.03 |
| | **RotatE** | 72.37 | 72.88 | 55.73 |
| | **R-GCN** | 37.06 | 89.01 | 79.19 |
| **MoE** | **TransE** | 42.42 | 60.62 | 63.78 |
| | **DistMult** | 42.37 | 61.05 | 71.64 |
| | **RotatE** | 25.73 | 58.37 | 62.15 |
| | **R-GCN** | 22.75 | 63.04 | 64.56 |
| **Q-former** | **TransE** | 77.20 | 62.73 | 31.73 |
| | **DistMult** | 37.04 | 89.93 | 40.80 |
| | **RotatE** | 39.84 | 91.96 | 22.83 |
| | **R-GCN** | 40.73 | 87.41 | 25.86 |

### B.3 The influence of textual query in TG-MC task.

As we presented in the task definitions of SUBARU, we provide text-based descriptions of the given query in the TG MC task. For example, *([MASK] | occupation | romanist)* in Figure 14. Besides, the options are also in the form of texts which means many questions can make text-based predictions without SKP as well. To better investigate LLM's ability to understand SKP on the task TG MC, we performed some additional implementations to validate the experiments in the absence of text. As shown in Figure 4, it is obvious that the model performs better in the presence of text, because the query present in the form of text greatly simplifies LLM's understanding of the problem and makes the whole thing easier. But on the other hand, in the absence of text, the LLM still has some SKP comprehension and it can make the right choices relying on SKP alone.

### B.4 Additional results on LlaMA3.1 and LlaMA2

We present more detailed experimental results in Figure 5 and Figure 6 about the SUBARU benchmark of LlaMA3.1-8B-Instruct and LlaMA2-7B-chat. These results are complementary to the universality experiment. We can observe that the model based on LlaMA3.1 performs relatively better in general compared to the model based on LlaMA2.

**Entity Classification (EG-CLS)**

<Structural Knowledge Prompting $\mathcal{S}_e$ >
### **Instruction**: Given an entity embedding in the front and answer the following question.
### **Question:** Is this entity Bonnie Owens?
### **Output:** Yes.

Figure 10: The prompt template for EG-CLS.

**Entity Multiple Choice (EG-MC)**

<Structural Knowledge Prompting $\mathcal{S}_e$ >
### **Instruction**: Given an entity embedding in the front and select the true answer.
### **Question:** Which is the possible type of entity? A. bell tower B. congestive heart failure C. academic major D. wall hanging
### **Output:** Your Answer is: D

Figure 11: The prompt template for EG-MC.

**Entity Description (EG-DESC)**

<Structural Knowledge Prompting $\mathcal{S}_e$ >
### **Instruction**: Given an entity embedding in the front and describe this entity.
### **Output:** Bonnie Owens. ### Dessctiption: Bonnie Owens , born Bonnie Campbell, was an American country music singer who was married to Buck Owens and later Merle Haggard.

Figure 12: The prompt template for EG-DESC.

**Triple Classification (TG-CLS)**

<Structural Knowledge Prompting $\mathcal{S}_h$, $\mathcal{S}_r$, $\mathcal{S}_t$ >
### **Instruction**: Given the embeddings of an knowledge triple in the front. Please answer the following question.
### **Question:** Is this a positive triple?
### **Output:** No

Figure 13: The prompt template for TG-CLS.

**Triple Multiple Choice (TG-MC)**

<Structural Knowledge Prompting $\mathcal{S}_r$, $\mathcal{S}_t$ > # Note that the masked answer does not appear in SKP
### **Instruction**: Given the embeddings of a query and four candidates in the front. Select a correct answer to fill the [MASK] and complete the triple.
### **Question:** ([MASK] | occupation | romanist) A. László András B. Rebekah Brooks C. Franz Konwitschny D. Francisco Rodríguez Marín
### **Output:** A

Figure 14: The prompt template for TG-MC.

**Triple Description (TG-DESC)**

<Structural Knowledge Prompting $\mathcal{S}_h, \mathcal{S}_r, \mathcal{S}_t$ >
### **Instruction**: Given the embeddings of an knowledge triple in the front and describe the head entity, relation, and tail entity of the triple.
### **Output:** Billy Idol###languages spoken, written, or signed###English

Figure 15: The prompt template for TG-DESC.

**Subgraph Classification (SG-CLS)**

<Structural Knowledge Prompting $\mathcal{S}_h, \mathcal{S}_{r_1}, \mathcal{S}_{t_1}, \cdots, \mathcal{S}_{r_k}, \mathcal{S}_{t_k}$>
### **Instruction**: Given embeddings from a subgraph in the front and answer the following question.
### **Question:** Is there any anomaly in this subgraph?
### **Output:** No

Figure 16: The prompt template for SG-CLS.

**Subgraph Multiple Choice (SG-MC)**

<Structural Knowledge Prompting $\mathcal{S}_{r_1}, \mathcal{S}_{t_1}, \cdots, \mathcal{S}_{r_k}, \mathcal{S}_{t_k}$> # The center entity does not appear.
### **Instruction**: Given an entity embedding in the front and select the true answer.
### **Question:** Which is the center entity descripted by this subgraph? A. The Lord of the Rings: The Fellowship of the Ring B. Christian Reimers C. Harry Fett D. Manuel Acevedo
### **Output:** Your Answer is: D

Figure 17: The prompt template for SG-MC.

**Subgraph Description (SG-DESC)**

<Structural Knowledge Prompting $\mathcal{S}_h, \mathcal{S}_{r_1}, \mathcal{S}_{t_1}, \cdots, \mathcal{S}_{r_k}, \mathcal{S}_{t_k}$>
### **Instruction**: Given embeddings from a subgraph in the front and answer the following question.
### **Output:** Dominic Howard is a musician from the United Kingdom, known for playing the drum kit in the genre of alternative rock. He primarily works in the field of music.

Figure 18: The prompt template for SG-DESC.