LLMs as Parametric Knowledge Sources for Knowledge Graph Completion

Deyu Chen[1] , **Qiyuan Li**[2,3] , **Jinguang Gu**[2,3] , **Meiyi Xie**[1] and **Hong Zhu**[1*]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]School of Computer Science and Technology, Wuhan University of Science and Technology, China
[3]Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, China
{deyuchen, xiemeiyi, zhuhong}@hust.edu.cn, {vickyuan, simon}@wust.edu.cn

Abstract

Knowledge graphs (KGs) serve as crucial symbolic knowledge sources for many downstream applications but often suffer from incompleteness. Recent efforts have attempted to integrate pre-trained language models and KGs for the knowledge graph completion (KGC) task. However, existing methods introduce substantial model complexity or rely on prompt-based solutions that inject priors at a shallow level, which do not exploit the factual knowledge inherently encoded in model parameters. To address these issues, we propose a novel framework that treats LLMs as parametric knowledge sources for KGC. Our framework performs LLM knowledge elicitation to extract factual knowledge from the model's internal representations. Then, a cross-granularity representation alignment method transforms sentence-level representations into entity-level representations and aligns them within a unified space. Finally, a dynamic learning schedule balances alignment and expressiveness throughout training. Extensive experiments on multiple benchmark datasets show that our proposed method can be integrated with diverse KGC baselines and consistently improves link prediction in both standard and lifelong settings.

1 Introduction

Knowledge graphs (KGs) are widely used to store factual knowledge as triples (subject, relation, object). By organizing entities and relations in a structured form, KGs support a variety of application scenarios such as recommender systems [Zhang *et al.*, 2024a], question answering [Ma *et al.*, 2025], and diagnostic assistance [Li *et al.*, 2025]. However, large real-world KGs are inevitably incomplete, which limits their effectiveness in downstream tasks. This makes knowledge graph completion (KGC) an essential task, aiming to predict missing facts based on existing triples.

The emergence of pre-trained language models (PLMs) and LLMs has significantly changed KG representation learning. Traditional KGC methods embed entities and relations into a continuous space to predict missing facts with geometric scoring functions. More recently, PLM-enhanced KGC approaches leverage the rich linguistic and world knowledge embedded in pre-trained models and can be broadly categorized into two paradigms. The first line of work treats KGs as textual corpora and fine-tunes PLMs/LLMs with KG data [Yao *et al.*, 2019; Lv *et al.*, 2022]. This can be effective, but it is often computationally expensive and may underutilize inherent structural information of KGs due to the inconsistency between language modeling and graph learning objectives. The second paradigm uses prompt design to introduce semantic priors from LLMs into the KGC learning process [Wei *et al.*, 2023; Jiang *et al.*, 2024]. Such methods typically provide only shallow semantic guidance and rely on surface-level prompts, leaving the internal parametric factual knowledge encoded within LLMs insufficiently utilized. Figure 1 illustrates these two paradigms and our proposed elicitation-based paradigm. Rather than adapting PLMs/LLMs to KG triples or relying on guidance from LLM

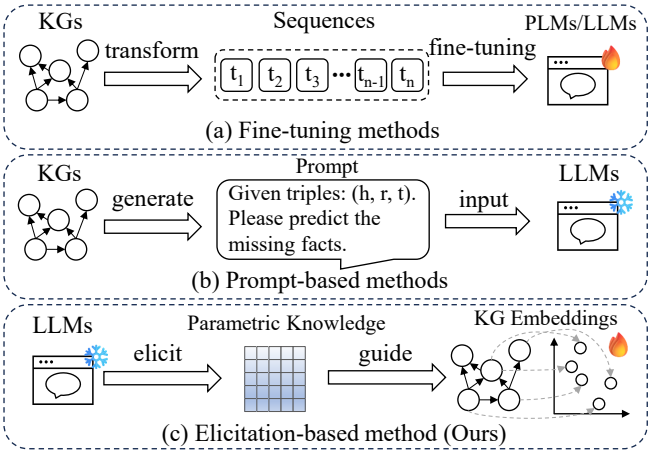

Figure 1: Three paradigms of PLM-enhanced KGC. (a) Fine-tuning methods convert triples into token sequences for fine-tuning PLMs/LLMs to produce predictions. (b) Prompt-based methods query LLMs via prompt design for responses. (c) Our elicitation-based method elicits factual representations from internal representations to guide the learning process of KG embeddings.

*Corresponding author.

textual outputs, LarKS treats LLMs as parametric knowledge sources and uses their internal representations to provide representation-level guidance for KGC.

Recent advances in understanding how factual knowledge is stored in Transformer-based language models [Geva *et al.*, 2021; Meng *et al.*, 2022; Hernandez *et al.*, 2024] have provided new insights into their internal representations. These studies suggest that PLMs implicitly encode a substantial amount of factual knowledge in their parameters. This observation motivates us to explore how such parametric knowledge can be effectively elicited and leveraged to enhance KGC. However, two critical challenges arise: First, factual knowledge may be implicitly stored across different layers and neurons of LLMs, rather than being directly accessible from their responses. Second, there is an objective mismatch between next-token prediction in LLMs and the embedding-based learning objective commonly used in KGC methods.

To address the above issues, we propose a novel PLM-enhanced KGC paradigm, LarKS, that leverages LLMs as parametric knowledge sources to enhance the KGC task. Our framework elicits factual representations from LLM internal representations and aligns them with the representation space learned by KGC models. Specifically, we introduce an LLM knowledge elicitation mechanism to extract representations from intermediate hidden states of existing pre-trained Transformer language models. To bridge the gap between the sentence-level representations of LLMs and the entity-level embeddings used in KGC, we design a cross-granularity representation alignment method that transforms and aligns them. Finally, to ensure stable and effective joint optimization, a dynamic learning schedule is adopted to balance the alignment and representation learning objectives throughout training.

The main contributions of our work are summarized as follows:

- We propose a novel learning paradigm for KGC methods, where LLMs are treated as parametric knowledge sources to enrich the representation learning of KGs. This paradigm provides a promising way to bridge the implicit parametric knowledge in LLMs and the explicit symbolic facts in KGs.
- We design a PLM-enhanced KGC framework, LarKS. It includes a knowledge elicitation mechanism for extracting factual representations from LLMs, a cross-granularity alignment method for bridging sentence-level and entity-level representations, and a dynamic learning schedule for balancing joint objectives.
- We conduct extensive experiments on several benchmark datasets under both standard and lifelong settings. The results demonstrate the effectiveness of our method and highlight the potential of integrating parametric knowledge into KGC methods.

2 Related Work

2.1 PLM-enhanced KGC

Traditional KGC methods learn vector representations of entities and relations, and define a scoring function over triples to predict missing facts. Prior work has explored increasingly expressive scoring functions, ranging from geometric transformations [Bordes *et al.*, 2013; Sun *et al.*, 2019] to factorized bilinear interactions [Yang *et al.*, 2015; Trouillon *et al.*, 2016], and further to convolutional interaction modules [Dettmers *et al.*, 2018].

The rapid development of PLMs/LLMs, which encode rich world knowledge and semantic priors, has motivated growing interest in combining PLMs with KGC. Existing PLM-enhanced KGC methods can be broadly grouped into two categories: (1) Fine-tuning methods convert KG triples and contexts into input sequences for PLMs/LLMs and optimize them for link prediction. KG-BERT [Yao *et al.*, 2019] and PKGC [Lv *et al.*, 2022] fine-tune PLMs for triple scoring. KoPA [Zhang *et al.*, 2024b], MKGL [Guo *et al.*, 2024], and SSQR [Lin *et al.*, 2025] further augment tuning with KG-derived structural information or specialized KG codes. However, fine-tuning methods are costly and easily disrupt KG structural information. (2) Prompt-based methods use designed prompts to obtain guidance from frozen LLMs. KICGPT [Wei *et al.*, 2023] and ARR [Chen *et al.*, 2024] use LLMs for candidate re-ranking. CSProm-KG-CD [Li *et al.*, 2024] distills LLMs' generated triple descriptions into smaller KGC models. KG-FIT [Jiang *et al.*, 2024] uses LLM-guided refinement to build a hierarchical structure of entity clusters for improving KGC training. Nevertheless, prompt-based methods provide shallow semantic augmentation and may produce hallucinated and underspecified results.

2.2 Factual Representations in Language Models

A growing line of research investigates how factual knowledge is stored and represented inside LLMs[1]. These studies primarily aim to explore the internal mechanisms through which Transformer language models encode factual associations and how such representations influence model predictions. Feed-forward layers in Transformers have been found to act as key–value memories encoding input patterns and their associated output distributions [Geva *et al.*, 2021; Geva *et al.*, 2022]. ROME [Meng *et al.*, 2022] reveals that factual knowledge is stored in highly causal "early-site" MLP activations at the last subject token, which is later leveraged by MEMIT [Meng *et al.*, 2023] to enable large-scale model updating. Further studies [Sakata *et al.*, 2025; Morand *et al.*, 2025] explore the finer-grained correlation between entities and their mentions within LLM hidden representations. LRE [Hernandez *et al.*, 2024] shows that many relational associations in TLMs exhibit an approximately affine mapping from subject to object representations. Inspired by these efforts, we explore whether such internal factual representations can be elicited and leveraged to enhance KGC.

3 Methodology

We propose LarKS, a PLM-enhanced KGC framework that leverages the parametric knowledge in pretrained LLMs for KG representation learning. As shown in Figure 2, LarKS

[1]Most analyses relied on Transformer language models (TLMs). Since LLMs predominantly adopt this architecture, we do not strictly distinguish between "TLM" and "LLM".

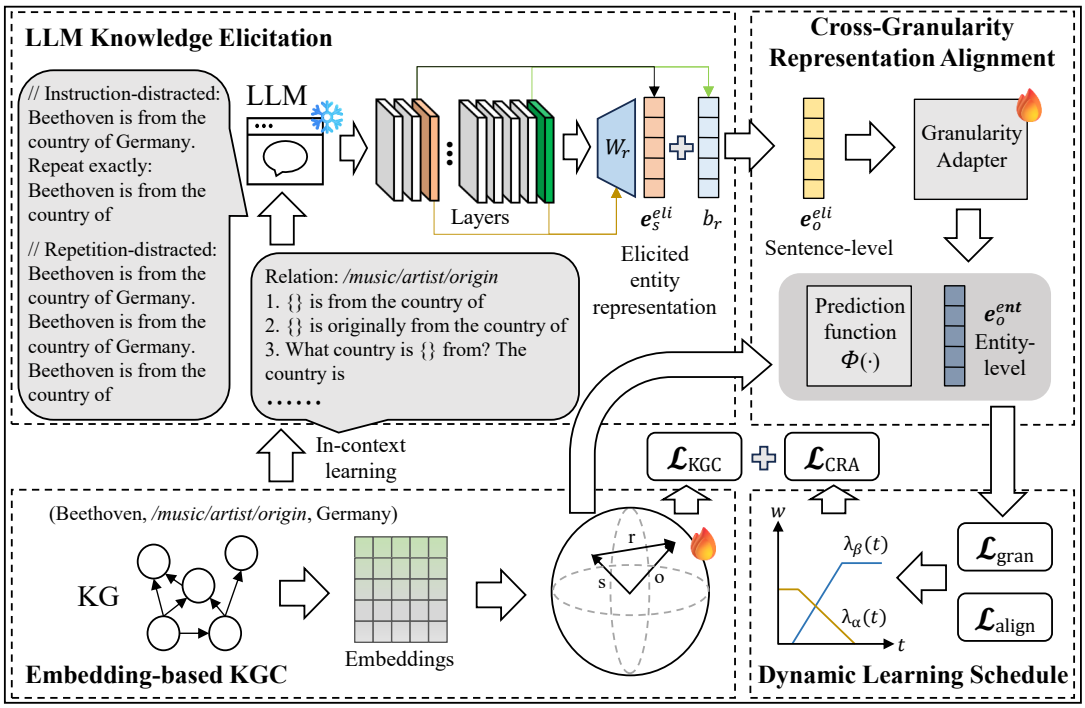

Figure 2: The overview of our proposed framework.

elicits factual representations from the LLM's internal representations, bridges the granularity mismatch to obtain entity-level guidance, and jointly optimizes objectives of different stages under a dynamic learning schedule.

3.1 LLM Knowledge Elicitation

The knowledge elicitation module extracts factual representations from pretrained LLMs' intermediate hidden states to obtain elicited entity representations and relational transformations, serving as prior semantic guidance from the LLM's parametric general knowledge. We design prompt templates to induce the LLM to produce the expected factual prediction, and take the corresponding hidden states as elicited representations for subsequent modules.

Elicited Representations Extraction

Recent works [Geva *et al.*, 2021; Meng *et al.*, 2022; Hernandez *et al.*, 2024] have explored how factual knowledge is stored within TLMs.

Building on causal intervention analysis [Meng *et al.*, 2022], prior work identifies strongly causal states at an early site in the middle layers of autoregressive TLMs. These states occur at the last subject token and have been shown to encode entity-level factual information that influences the model's predictions. Motivated by this, we extract elicited entity representations from the early-site hidden state of the last subject token. Specifically, given an input sequence of length T, a TLM f_θ with L layers produces layer-wise hidden states $\mathbf{H}^{(l)} = f_\theta^{(l)}(\mathbf{H}^{(l-1)})$ for $l = 1, \ldots, L$, where $\mathbf{H}^{(l)} = [\mathbf{h}_1^{(l)}; \ldots; \mathbf{h}_T^{(l)}] \in \mathbb{R}^{T \times d}$ and d is the hidden size. For a subject entity e tokenized as m tokens with its last token position p_m, we select the hidden state of the last subject token from an early layer l_n (typically $l_n \leq L/2$) as its elicited entity representation:

$$\mathbf{e} = \mathbf{h}_{p_m}^{(l_n)}, \tag{1}$$

where $\mathbf{e}$ denotes the elicited entity representation of e. To maintain consistency in elicited entity representations, we use a simple fixed prompt template, `{} is a ...`, where `{}` is replaced with the entity name, which reduces context-dependent variation and yields hidden states that primarily reflect entity-level semantics. We elicit entity representations and relation transformations separately to decouple their estimation dependencies and better align with embedding-based KGC methods.

Our relation extraction is grounded in the linear relational embedding (LRE) [Hernandez *et al.*, 2024], which shows that, for a subset of relations, the transformation from subject to object representations can be well-approximated by a single relation-specific affine mapping. Formally, given a relation-specific prompt context c_r, the induced mapping can be written as: $\mathbf{o} = F(\mathbf{s}, c_r)$, where $\mathbf{s}$ is the intermediate-layer hidden representation at the last subject token and $\mathbf{o}$ is the hidden state used to decode the object's first token. We approximate F with a relation-specific affine transformation:

$$F(\mathbf{s}, c_r) \approx \lambda_{\text{corr}} W_r \mathbf{s} + b_r, \tag{2}$$

where W_r and b_r are the relation-specific transformation matrix and bias term, and $\lambda_{\text{corr}}\,(> 1)$ is a scalar constant that corrects the magnitude underestimation caused by layer normalization. Following LRE, we use a first-order approximation with Jacobian $J(\mathbf{s}, c) = \partial F(\mathbf{s}, c)/\partial \mathbf{s}$, and estimate W_r

and b_r by taking expectations over examples $(\mathbf{s}_i, c_i)$ of relation r:

$$W_r = \mathbb{E}_{(\mathbf{s}_i, c_i)\sim r}[J(\mathbf{s}_i, c_i)]\,, \tag{3}$$

$$b_r = \mathbb{E}_{(\mathbf{s}_i, c_i)\sim r}[F(\mathbf{s}_i, c_i) - J(\mathbf{s}_i, c_i)\mathbf{s}_i]\,. \tag{4}$$

In practice, we approximate the expectations by averaging over the sampled examples.

Prompt Design and Guidance Strategy

Since large-scale KGs contain many different relations, we design a prompt template generation strategy based on in-context learning, avoiding manual prompt template design for each new relation. Here, prompts are used to control the elicitation of internal representations and relation transformations, rather than to query the LLM for final KGC prediction. Specifically, we adopt a set of seed relations with well-crafted prompt templates, containing several typical relation categories. For an unseen relation, we encode its relation text and each seed relation text with an embedding model, compute their semantic similarity in the embedding space, and use prompt templates of top-M seed relations as in-context demonstrations. With these demonstrations, the LLM generates N candidate prompt templates for the new relation. Figure 2 provides an example of the generated templates for the relation */music/artist/origin*. The generated prompt templates provide multiple prompt patterns for the same relation, enabling parameter estimation from diverse contexts and yielding more stable results.

However, this process cannot ensure the model produces factual predictions, and incorrect outputs will distort the estimated relation transformations. To address this issue, we adopt distracted prompts as model inputs, which induce the model to produce the expected response while having no significant impact on the resulting estimates [Hernandez *et al.*, 2024]. We first apply the instruction-distracted form, and use the repetition-distracted form as a fallback when the model still fails to produce the expected prediction. Figure 2 also illustrates the two distracted prompt designs for the triple $(\textit{Beethoven}, \textit{/music/artist/origin}, \textit{Germany})$. Empirically, this two-stage prompting strategy suffices for all samples in our experiments. For rare cases where the model still produces an incorrect prediction, which typically correspond to long-tail or difficult samples, we resample a new instance instead. These designs help stabilize the elicitation process for relation transformation estimation.

3.2 Cross-Granularity Representation Alignment

A key challenge in PLM-enhanced KGC is the misalignment between sentence-level representations from LLMs and entity-level embeddings learned by KGC methods. This granularity discrepancy prevents these representations from effectively guiding KGC optimization. We address this issue with cross-granularity representation alignment, combining a granularity transformation module with a representation alignment module.

Granularity Transformation

LLMs trained under the next-token prediction objective primarily yield sentence-level contextual representations, which are mismatched with the entity-level embeddings learned by KGC methods. Recent work on the injectivity and invertibility of language models suggests that hidden representations can retain recoverable information about input sequences [Nikolaou *et al.*, 2025]. This motivates us to treat the granularity gap as a learnable transformation problem. Specifically, we introduce a lightweight granularity adapter to convert sentence-level representations into entity-level embeddings. For a triple (s, r, o), we employ a layer normalization followed by a linear transformation to stabilize representations and perform granularity transformation:

$$\mathbf{e}_o^{\text{ent}} = W_{\text{gran}}\,\mathrm{LN}(\mathbf{e}_o^{\text{sent}}) + b_{\text{gran}}, \tag{5}$$

where $\mathbf{e}_o^{\text{sent}}$ is the sentence-level representation of o obtained by transforming the elicited subject representation $\mathbf{e}_s$ with the relation transformation in Eqs. 3 and 4, $\mathbf{e}_o^{\text{ent}}$ is the corresponding entity-level representation produced by the granularity adapter, W_{gran} and b_{gran} are learnable parameters, and $\mathrm{LN}(\cdot)$ denotes layer normalization.

To enable elicited representations to better guide embedding-based KGC training, we encourage consistency between the predictive distributions produced by the granularity adapter and the embedding-based KGC method:

$$\mathcal{L}_{\text{gran}} = \mathrm{KL}\left(P_{\text{kgc}}(\cdot \mid s, r) \parallel P_{\text{llm}}(\cdot \mid s, r)\right), \tag{6}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence, which promotes adaptation of the LLM-derived predictions to the granularity and semantic space of the KGC method under joint optimization. We construct the two distributions with the elicited entity representation matrix $\mathbf{E}^{\text{eli}}$ and the entity embedding matrix $\mathbf{E}^{\text{kgc}}$ of the KGC method:

$$P_{\text{llm}}(\cdot \mid s, r) = \mathrm{softmax}\big(\mathbf{e}_o^{\text{ent}}(\mathbf{E}^{\text{eli}})^\top\big), \tag{7}$$

$$P_{\text{kgc}}(\cdot \mid s, r) = \mathrm{softmax}\big(\phi(\mathbf{e}_s^{\text{kgc}}, \mathbf{r})(\mathbf{E}^{\text{kgc}})^\top\big), \tag{8}$$

where $\mathbf{r}$ denotes the relation embedding in the KGC method, and $\phi(\cdot)$ is the entity–relation composition function that composes the subject and relation embeddings into a predicted object representation for link prediction. This composition is consistent with the scoring function of the KGC method. For example, TransE uses the score $f(s, r, o) = -\|\mathbf{e}_s^{\text{kgc}} + \mathbf{r} - \mathbf{e}_o^{\text{kgc}}\|$, and thus its composition function is $\phi(\mathbf{e}_s^{\text{kgc}}, \mathbf{r}) = \mathbf{e}_s^{\text{kgc}} + \mathbf{r}$.

Representation Alignment

We further enforce their consistency with a representation alignment objective, complementing distribution-level guidance. This alignment mechanism encourages representations of the same entity to be close and suppresses incorrect ones, consistent with CLIP-style representation learning [Radford *et al.*, 2021; Brannon *et al.*, 2024; Zhu *et al.*, 2025]. Accordingly, we measure the distance between the transformed entity representation and the predicted object representation produced by the KGC method as follows:

$$d_{\text{align}}(s, r, o) = \|\mathbf{e}_o^{\text{ent}} - \phi(\mathbf{e}_s^{\text{kgc}}, \mathbf{r})\|. \tag{9}$$

To optimize the distance with negative triples, we adopt the widely used negative sampling loss [Mikolov *et al.*, 2013]:

$$\begin{aligned}\mathcal{L}_{\text{align}} = &-\log\sigma\big(\gamma - d_{\text{align}}(s, r, o)\big) \\ &- \sum_{i=1}^{n}\frac{1}{n}\log\sigma\big(d_{\text{align}}(s_i', r, o_i') - \gamma\big),\end{aligned} \tag{10}$$

where γ is the margin hyperparameter, $\sigma(\cdot)$ is the sigmoid function, n is the number of negative samples, and (s'_i, r, o'_i) denotes the i-th negative sample corresponding to (s, r, o). For prediction, the final representation is obtained by a weighted fusion: $\mathbf{e}_o = \eta\phi(\mathbf{e}_s^{\text{kgc}}, \mathbf{r}) + (1-\eta)\mathbf{e}_o^{\text{ent}}$, where $\eta \in [0, 1]$ is a hyperparameter.

3.3 Dynamic Learning Schedule

During the joint optimization of LLM elicited representations and KGC methods, the optimization focus evolves over different training stages. In the early stage of training, we emphasize granularity transformation to make the two representations comparable. This reduces the risk of suboptimal local minima early on, allowing both sides to develop useful structures before stronger semantic alignment. As training proceeds, once the two representations become comparable in granularity and distribution, the optimization focus should shift toward representation alignment. This stage enforces semantic consistency by bringing representations of the same entity closer and suppressing mismatched ones, thereby enabling effective knowledge sharing and mutual reinforcement.

To this end, we adopt a dynamic learning schedule that gradually shifts the training emphasis across stages. Specifically, we introduce two time-varying weights to balance the contributions of granularity transformation and representation alignment:

$$\mathcal{L}_{\text{CRA}}(t) = \lambda_\alpha(t)\,\mathcal{L}_{\text{gran}} + \lambda_\beta(t)\,\mathcal{L}_{\text{align}}, \tag{11}$$

where t denotes the current training step. In practice, we observed that a simple clipped linear schedule can be effective as follows:

$$\lambda(t) = w^{\text{s}} + \text{clip}\left(\frac{t - t_{\text{s}}}{t_{\text{e}} - t_{\text{s}}}, 0, 1\right)\left(w^{\text{e}} - w^{\text{s}}\right), \tag{12}$$

where t_{s} and t_{e} denote the start and end epochs, while w^{s} and w^{e} are the corresponding initial and final weights. Here, $\text{clip}(x, 0, 1) = \max\big(0, \min(1, x)\big)$. The weight of granularity transformation $\lambda_\alpha(t)$ decreases linearly as training proceeds, while the weight of representation alignment $\lambda_\beta(t)$ increases, gradually shifting the training emphasis from granularity reconciliation to representation consistency.

3.4 Total Training Objective

Our overall training objective integrates the above modules for joint optimization. The total loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{KGC}} + \mathcal{L}_{\text{CRA}}, \tag{13}$$

where $\mathcal{L}_{\text{KGC}}$ adopts the same form as Eq. 10, with the model-specific distance term defined as the negative score $d(s, r, o) = -f(s, r, o)$. LarKS jointly leverages KGC representations and elicited representations from LLMs, allowing them to complement each other during training.

4 Experiments

4.1 Experimental Settings

We evaluate our framework in two settings: standard KGC and lifelong learning [Cui *et al.*, 2023]. For the latter, we leverage LLMs' broad world knowledge and generalization ability to continually learn from evolving KG snapshots while avoiding forgetting previously acquired facts.

Dataset	Entities	Relations	# Train	# Valid	# Test
FB15k-237	14,541	237	272,115	17,535	20,466
FB15k-237N	13,104	93	87,282	7,041	8,226
Wiki27k	27,122	62	74,793	10,121	10,122

Table 1: Statistics of benchmark datasets for standard KGC.

Datasets. For standard KGC, we use three benchmark datasets, FB15k-237 [Toutanova and Chen, 2015], FB15k-237N [Lv *et al.*, 2022], and Wiki27k [Lv *et al.*, 2022], sampled from Freebase and Wikidata. Table 1 reports the statistics of these standard KGC datasets. For the lifelong setting, we adopt seven benchmark datasets constructed under different incremental strategies (ENTITY, RELATION, FACT, HYBRID) and speeds (GraphEqual, GraphHigher, GraphLower), following previous works [Cui *et al.*, 2023; Liu *et al.*, 2024a].

Baselines. In the standard KGC, we compare our method with: (1) Traditional embedding-based methods: TransE [Bordes *et al.*, 2013], DistMult [Yang *et al.*, 2015], ComplEx [Trouillon *et al.*, 2016], ConvE [Dettmers *et al.*, 2018], and RotatE [Sun *et al.*, 2019]; (2) PLM-enhanced methods: KG-BERT [Yao *et al.*, 2019], PKGC [Lv *et al.*, 2022], and KG-FIT [Jiang *et al.*, 2024]. For the lifelong setting, we adopt twelve baselines including: (1) Non-continual methods: Snapshot and Fine-tune; (2) Dynamic architecture-based methods: PNN [Rusu *et al.*, 2016], CWR [Lomonaco and Maltoni, 2017], and FastKGE [Liu *et al.*, 2024b]; (3) Rehearsal-based methods: GEM [Lopez-Paz and Ranzato, 2017], EMR [Wang *et al.*, 2019], and DiCGRL [Kou *et al.*, 2020]; (4) Regularization-based methods: EWC [Kirkpatrick *et al.*, 2017], SI [Zenke *et al.*, 2017], LKGE [Cui *et al.*, 2023], and IncDE [Liu *et al.*, 2024a].

Evaluation Metrics. Following common practice, we evaluate link prediction using mean reciprocal rank (MRR) and Hits@N (H@N) under the filtered setting [Bordes *et al.*, 2013]. In the lifelong setting, we evaluate the final model on the union of all test sets, restricting candidate entities of each query to those observed up to its snapshot.

Implementation Details. All experiments are implemented in PyTorch [Paszke *et al.*, 2019], and training uses the Adam optimizer [Kingma and Ba, 2015]. We follow prior implementations [Lv *et al.*, 2022; Jiang *et al.*, 2024; Liu *et al.*, 2024a] and their main settings when applicable. We tune the batch size in $\{512, 1024, 2048\}$, and set the hidden dimension to 1024 in the standard setting and 200 in the lifelong setting. For LLM components, we use Qwen3-8B (default) and Llama 3.1-8B (analysis in Section 4.4) as backbones, and adopt Qwen3-Embedding-4B (prompt template generation in Section 3.1) and OpenAI text-embedding-3-large (ablation variant in Section 4.3) as embedding models. During training, we evaluate validation MRR every 3 epochs and apply early stopping with patience 5. For step-based runs, we treat one dataset round as one epoch.

	Model	FB15k-237			FB15k-237N			Wiki27k		
		MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
Traditional	TransE	0.287	0.192	0.478	0.304	0.213	0.474	0.155	0.032	0.378
	LarKS (TransE)	0.325	0.226	0.522	0.328	0.232	0.509	0.181	0.060	0.419
	DistMult	0.260	0.163	0.437	0.277	0.197	0.431	0.218	0.130	0.405
	LarKS (DistMult)	0.300	0.214	0.472	0.283	0.204	0.432	0.224	0.138	0.402
	ComplEx	0.252	0.161	0.439	0.261	0.176	0.428	0.230	0.153	0.394
	LarKS (ComplEx)	0.309	0.220	0.492	0.296	0.216	0.454	0.246	0.167	0.414
	ConvE	0.312	0.224	0.508	0.273	0.192	0.429	0.226	0.164	0.354
	LarKS (ConvE)	0.320	0.233	0.502	0.283	0.206	0.434	0.235	0.173	0.361
	RotatE	0.333	0.241	0.528	0.308	0.226	0.465	0.216	0.123	0.394
	LarKS (RotatE)	0.339	0.247	0.532	0.333	0.245	0.503	0.237	0.149	0.399
PLM-enhanced	KG-BERT	0.245	0.158	0.420	0.203	0.139	0.403	0.192	0.119	0.352
	PKGC	0.342	0.236	0.525	0.332	**0.261**	0.487	0.285	**0.230**	0.409
	KG-FIT	0.349	0.250	0.545	0.347	0.246	0.538	0.271	0.189	0.429
	LarKS (KG-FIT)	**0.360**	**0.261**	**0.553**	**0.361**	0.258	**0.561**	**0.287**	0.205	**0.458**

Table 2: Link prediction results in the standard setting on FB15k-237, FB15k-237N, and Wiki27k. **Bold** numbers denote the best results, and underlined numbers denote the second best results.

Model	ENTITY		RELATION		FACT		HYBRID		GraphEqual		GraphHigher		GraphLower	
	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
Non-continual Learning Methods														
Snapshot	0.084	0.193	0.021	0.043	0.082	0.191	0.036	0.077	0.127	0.283	0.189	0.373	0.036	0.089
Fine-tune	0.165	0.321	0.093	0.195	0.172	0.339	0.135	0.262	0.183	0.358	0.198	0.375	0.185	0.363
Dynamic Architecture-based Methods														
PNN	0.229	0.425	0.167	0.305	0.157	0.290	0.185	0.349	0.212	0.405	0.186	0.364	0.213	0.407
CWR	0.088	0.202	0.021	0.043	0.083	0.192	0.037	0.077	0.122	0.277	0.189	0.374	0.032	0.080
FastKGE	0.239	0.427	0.185	0.359	0.203	0.384	0.211	0.382	0.217	0.398	0.201	0.365	0.218	0.402
Rehearsal-based Methods														
GEM	0.165	0.321	0.093	0.196	0.175	0.345	0.136	0.263	0.189	0.372	0.197	0.372	0.170	0.346
EMR	0.171	0.330	0.111	0.225	0.171	0.337	0.141	0.267	0.185	0.359	0.202	0.379	0.188	0.362
DiCGRL	0.107	0.211	0.133	0.241	0.162	0.320	0.149	0.277	0.104	0.226	0.116	0.242	0.102	0.222
Regularization-based Methods														
EWC	0.229	0.423	0.165	0.306	0.201	0.382	0.186	0.350	0.207	0.400	0.198	0.385	0.210	0.405
SI	0.154	0.311	0.113	0.224	0.172	0.343	0.111	0.229	0.179	0.353	0.190	0.371	0.186	0.366
LKGE	0.234	0.425	0.192	0.366	0.210	0.387	0.207	0.379	0.214	0.407	0.207	0.382	0.210	0.403
IncDE	0.253	0.448	0.199	0.370	**0.216**	**0.391**	**0.224**	**0.401**	0.234	0.432	**0.227**	**0.412**	0.228	0.426
LarKS (LKGE)	**0.262**	**0.480**	**0.207**	**0.374**	0.209	0.368	0.219	0.388	**0.237**	**0.447**	0.219	0.407	**0.233**	**0.437**

Table 3: Link prediction results in the lifelong setting on seven benchmark datasets: ENTITY, RELATION, FACT, HYBRID, GraphEqual, GraphHigher, and GraphLower. **Bold** numbers denote the best results, and underlined numbers denote the second best results.

4.2 Main Results

Standard Setting. Table 2 shows results on three standard KGC datasets. Overall, LarKS consistently improves over several baselines and achieves the best performance when built upon KG-FIT. For traditional methods, LarKS improves MRR over TransE, DistMult, ComplEx, ConvE, and RotatE by 12.6%, 6.8%, 14.3%, 3.4%, and 6.5%, respectively. On FB15k-237, FB15k-237N, and Wiki27k, LarKS improves MRR with average relative improvements of 11.1%, 7.1%, and 8.0%, respectively. These results demonstrate the effectiveness and adaptability of our proposed method. For PLM-enhanced baselines, LarKS (KG-FIT) achieves average relative MRR improvements of 18.5%, 30.2%, and 18.7% on FB15k-237, FB15k-237N, and Wiki27k, respectively. Compared with KG-FIT, our extended framework yields an average relative MRR gain of 4.4% across the three datasets. We observe that PKGC achieves higher Hits@1 on FB15k-237N and Wiki27k, since we report its definition-augmented variant that injects support prompts and fine-tunes a PLM-based classifier, which yields a clear Hits@1 improvement over the base PKGC. This suggests that fine-tuning PLMs with support prompts can provide a more targeted boost to semantic prediction, but requires higher training and computation costs. Moreover, KG-FIT has been shown to complement PKGC in a recall and re-rank pipeline [Jiang *et al.*, 2024], indicating that LarKS not only achieves better overall performance but also remains promising and extensible to such hybrid architectures.

Lifelong Setting. We integrate LarKS into the representative lifelong baseline LKGE for evaluation and report the results in Table 3. Overall, LarKS achieves the best average performance across the seven benchmarks. Compared

Model		FB15k-237		FB15k-237N		Wiki27k	
		MRR	H@10	MRR	H@10	MRR	H@10
TransE	w/o GT	0.313	0.510	0.319	0.501	0.177	0.412
	w/o RA	0.315	0.511	0.316	0.503	0.168	0.398
	w/o DLS	0.316	0.511	0.317	0.497	0.170	0.394
	w/ Emb	0.315	0.515	0.313	0.506	0.165	0.401
	LarKS	**0.325**	**0.522**	**0.328**	**0.509**	**0.181**	**0.419**
RotatE	w/o GT	0.333	0.526	0.325	0.492	0.223	0.391
	w/o RA	0.330	0.525	0.322	0.490	0.227	0.395
	w/o DLS	0.329	0.522	0.324	0.495	0.206	0.380
	w/ Emb	0.326	0.523	0.324	0.498	0.213	0.385
	LarKS	**0.339**	**0.532**	**0.333**	**0.503**	**0.237**	**0.399**

Table 4: Ablation study of LarKS. **Bold** numbers denote the best results, and underlined numbers denote the second best results.

with Snapshot and Fine-tune, it yields substantial gains, indicating effective continual learning capability under evolving snapshots. Compared with ten continual learning baselines, LarKS achieves average relative improvements of 62.6% and 48.6% in MRR and H@10, respectively. By integrating LarKS into LKGE, we obtain average relative improvements of 7.5% in MRR and 5.3% in H@10 over the original LKGE, while achieving overall performance comparable to the state-of-the-art baseline IncDE, without introducing any additional lifelong-specific components. Although LarKS is not the best on a few metrics, such as MRR on FACT, it does not incorporate additional designs specialized for the lifelong scenario. Nevertheless, the results still show that LarKS can effectively enhance KGC in the lifelong setting, benefiting from the expressive and transferable parametric knowledge of the LLM.

4.3 Ablation Study

To evaluate the contribution of each component in LarKS, we construct four variants: (1) w/o GT removes the granularity transformation module, (2) w/o RA removes the representation alignment module, (3) w/o DLS disables the dynamic learning schedule, and (4) w/ Emb replaces our elicited entity representations with embeddings obtained from entity texts/descriptions via OpenAI *text-embedding-3-large* and regularizes the model to stay close to these embeddings using a distance-based constraint. As shown in Table 4, LarKS yields 4.4% and 2.5% relative improvements in MRR and H@10 over the three removal variants. The improvements are averaged across three datasets and both TransE and RotatE, demonstrating the effectiveness and necessity of each module. Moreover, compared with w/ Emb, LarKS further improves MRR and H@10 by 5.9% and 2.1% on average, indicating that our elicited representations are more effective than directly relying on LLM-based embedding models.

4.4 Further Analysis

Effect of Elicitation Layers. We investigate how the choice of elicitation layer affects LarKS. Figure 3 shows the MRR of TransE and RotatE on FB15k-237 and Wiki27k when extracting LLM internal representations from four representative layers. Overall, eliciting from earlier layers consistently yields the best results. On FB15k-237, the performance peaks at layer 5 (our default setting) for both TransE and RotatE and tends to decrease as we move to deeper layers,

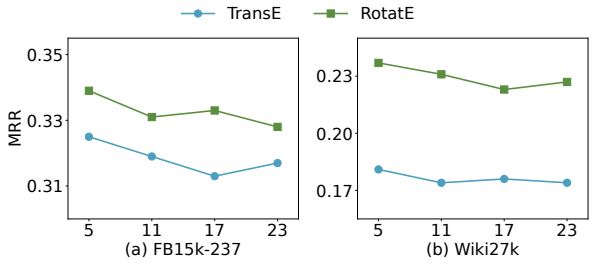

Figure 3: Results with different elicitation layers.

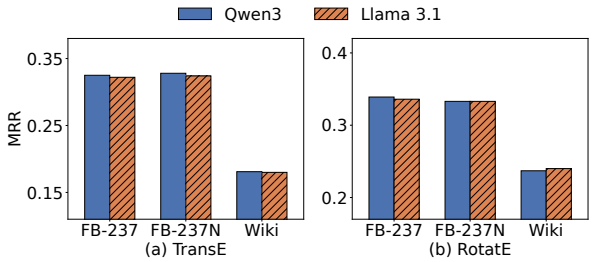

Figure 4: Results with different backbone LLMs.

while a similar trend is observed on Wiki27k. Although the performance is not strictly decreasing in all cases, the early layer setting is consistently superior. This suggests that the guidance leveraged by LarKS is more effective in early internal states, which preserve more localized semantics and are less affected by deeper layer contextualization.

Effect of Backbone LLMs. We further evaluate the effect of backbone LLMs on LarKS. Figure 4 shows the MRR of TransE and RotatE on FB15k-237, FB15k-237N, and Wiki27k when using Qwen3-8B (our default setting) and Llama 3.1-8B to elicit internal representations. LarKS achieves comparable performance across the two backbones. On FB15k-237 and FB15k-237N, Qwen3 yields slightly higher MRR for TransE and achieves comparable results to Llama 3.1 for RotatE. On Wiki27k, the two backbones perform similarly, with Llama 3.1 being marginally better for RotatE. These results indicate that, under comparable model scale, LarKS is not overly sensitive to backbone architecture details, and can effectively leverage parametric knowledge from different LLMs to provide guidance for KGC training.

5 Conclusion

In this paper, we propose a new learning paradigm for PLM-enhanced KGC that treats LLMs as parametric knowledge sources. Instead of fine-tuning or prompting LLMs, we elicit factual representations from their internal states and use them to guide the training of KGC methods. This design avoids the cost of LLM updating and shallow prompt-level usage, while leveraging the parametric knowledge implicitly encoded in the model to improve link prediction. Extensive experiments across multiple benchmarks in both standard and lifelong settings demonstrate that our framework consistently improves performance with different KGC methods.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on NIPS*, pages 2787–2795, 2013.

[Brannon *et al.*, 2024] William Brannon, Wonjune Kang, Suyash Fulay, Hang Jiang, Brandon Roy, Deb Roy, and Jad Kabbara. ConGraT: Self-supervised contrastive pretraining for joint graph and text embeddings. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 19–39, 2024.

[Chen *et al.*, 2024] Zhongwu Chen, Long Bai, Zixuan Li, Zhen Huang, Xiaolong Jin, and Yong Dou. A new pipeline for knowledge graph reasoning enhanced by large language models without fine-tuning. In *Proceedings of the 2024 Conference on EMNLP*, pages 1366–1381, 2024.

[Cui *et al.*, 2023] Yuanning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. Lifelong embedding learning and transfer for growing knowledge graphs. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 4217–4224, 2023.

[Dettmers *et al.*, 2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1811–1818. AAAI Press, 2018.

[Geva *et al.*, 2021] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[Geva *et al.*, 2022] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, 2022.

[Guo *et al.*, 2024] 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. MKGL: mastery of a three-word language. In *Proceedings of the Annual Conference on NIPS 2024*, 2024.

[Hernandez *et al.*, 2024] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *Proceedings of the Twelfth ICLR*. OpenReview.net, 2024.

[Jiang *et al.*, 2024] Pengcheng Jiang, Lang Cao, Cao (Danica) Xiao, Parminder Bhatia, Jimeng Sun, and Jiawei Han. KG-FIT: knowledge graph fine-tuning upon open-world knowledge. In *Proceedings of the Annual Conference on NIPS 2024*, 2024.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the 3rd ICLR*, 2015.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[Kou *et al.*, 2020] Xiaoyu Kou, Yankai Lin, Shaobo Liu, Peng Li, Jie Zhou, and Yan Zhang. Disentangle-based continual graph representation learning. In *Proceedings of the 2020 Conference on EMNLP*, pages 2961–2972, 2020.

[Li *et al.*, 2024] Dawei Li, Zhen Tan, Tianlong Chen, and Huan Liu. Contextualization distillation from large language model for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 458–477, 2024.

[Li *et al.*, 2025] Qiyuan Li, Haijiang Liu, Caicai Guo, Chao Gao, Deyu Chen, Meng Wang, Feng Gao, Frank van Harmelen, and Jinguang Gu. Reviewing clinical knowledge in medical large language models: Training and beyond. *Knowl. Based Syst.*, 328:114215, 2025.

[Lin *et al.*, 2025] Qika Lin, Tianzhe Zhao, Kai He, Zhen Peng, Fangzhi Xu, Ling Huang, Jingying Ma, and Mengling Feng. Self-supervised quantized representation for seamlessly integrating knowledge graphs with large language models. *CoRR*, abs/2501.18119, 2025.

[Liu *et al.*, 2024a] Jiajun Liu, Wenjun Ke, Peng Wang, Ziyu Shang, Jinhua Gao, Guozheng Li, Ke Ji, and Yanhe Liu. Towards continual knowledge graph embedding via incremental distillation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 8759–8768, 2024.

[Liu *et al.*, 2024b] Jiajun Liu, Wenjun Ke, Peng Wang, Jiahao Wang, Jinhua Gao, Ziyu Shang, Guozheng Li, Zijie Xu, Ke Ji, and Yining Li. Fast and continual knowledge graph embedding via incremental lora. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 2198–2206, 2024.

[Lomonaco and Maltoni, 2017] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pages 17–26, 2017.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the NIPS 2017*, pages 6467–6476, 2017.

[Lv *et al.*, 2022] Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. Do pre-trained models benefit knowledge graph completion? A reliable evaluation and a reasonable approach. In *Findings*

of the Association for Computational Linguistics, pages 3570–3581, 2022.

[Ma *et al.*, 2025] Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 24578–24597. Association for Computational Linguistics, 2025.

[Meng *et al.*, 2022] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Proceedings of the Annual Conference on NIPS 2022*, 2022.

[Meng *et al.*, 2023] Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *Proceedings of the Eleventh International Conference on Learning Representations*. OpenReview.net, 2023.

[Mikolov *et al.*, 2013] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th NIPS*, pages 3111–3119, 2013.

[Morand *et al.*, 2025] Victor Morand, Josiane Mothe, and Benjamin Piwowarski. On the representations of entities in auto-regressive large language models. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 433–451, 2025.

[Nikolaou *et al.*, 2025] Giorgos Nikolaou, Tommaso Mencattini, Donato Crisostomi, Andrea Santilli, Yannis Panagakis, and Emanuele Rodola. Language models are injective and hence invertible. *arXiv preprint arXiv:2510.15511*, 2025.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Annual Conference on NIPS 2019*, 2019.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th ICML*, volume 139, pages 8748–8763, 2021.

[Rusu *et al.*, 2016] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[Sakata *et al.*, 2025] Masaki Sakata, Benjamin Heinzerling, Sho Yokoi, Takumi Ito, and Kentaro Inui. On entity identification in language models. In *Findings of the Association for Computational Linguistics*, pages 16717–16741, 2025.

[Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th ICLR*, 2019.

[Toutanova and Chen, 2015] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33nd ICML*, volume 48, 2016.

[Wang *et al.*, 2019] Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 796–806, 2019.

[Wei *et al.*, 2023] Yanbin Wei, Qiushi Huang, Yu Zhang, and James T. Kwok. KICGPT: large language model with knowledge in context for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8667–8683, 2023.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd ICLR*, 2015.

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

[Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th ICML*, volume 70, pages 3987–3995, 2017.

[Zhang *et al.*, 2024a] Jin-Cheng Zhang, Azlan Mohd Zain, Kai-Qing Zhou, Xi Chen, and Ren-Min Zhang. A review of recommender systems based on knowledge graph embedding. *Expert Syst. Appl.*, 250:123876, 2024.

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

[Zhu *et al.*, 2025] Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pages 2183–2197. ACM, 2025.