TGCA-LLM: Time-Aware Graph-Text Contrastive Alignment for Enhancing LLMs in Temporal Knowledge Graph Completion

Zexuan Wan, Bo Wang, Kuofei Fang, Bin Wu*

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
School of Computer Science (National Pilot Software Engineering School),
Beijing University of Posts and Telecommunications, Beijing 100876, China
{wanzexuan, wangbo, coffeyf, wubin}@bupt.edu.cn

Abstract

Temporal Knowledge Graph Completion (TKGC) aims to infer missing facts by modeling historical events and latent temporal dependencies in Temporal Knowledge Graphs (TKGs). Recently, TKGC methods that integrate graph embeddings into Large Language Models (LLMs) have shown great promise by leveraging the structural information of TKGs together with the powerful reasoning capabilities of LLMs. However, these embedding-based methods are limited by suboptimal graph representations due to noise and long-tail issues in real-world scenarios, and insufficient cross-modal alignment between graph and language, hindering LLMs' ability to fully capture the temporal and structural information of TKGs. To address these issues, we propose TGCA-LLM, a novel embedding-based framework for TKGC. Specifically, TGCA-LLM first employs time-aware contrastive learning to align fact texts with graph structures in the temporal dimension, generating robust graph embeddings and establishing initial cross-modal alignment. Then, through a two-stage tuning process, it enables LLMs to gradually acquire structural and temporal knowledge from graph embeddings while enhancing their cross-modal reasoning capabilities in TKGC. Extensive experiments on three widely used real-world benchmarks demonstrate that TGCA-LLM outperforms state-of-the-art (SOTA) baselines by at least 8.7% MRR, highlighting its effectiveness.

Introduction

Knowledge Graphs (KGs) are structured representations of entities and relations that support various applications such as question answering (Liu et al. 2023), retrieval-augmented generation (Feng, Zhang, and Fei 2023; Sun et al. 2024), and recommendation systems (Qin et al. 2023). However, static KGs struggle to capture temporal evolution that are crucial for real-world reasoning. Therefore, Temporal Knowledge Graphs (TKGs) have drawn increasing attention (Han et al. 2021; Li, Sun, and Zhao 2022; Lee et al. 2023a), which represent time-sensitive facts as quadruples (*subject*, *relation*, *object*, *timestamp*). Although large-scale TKGs like ICEWS (Boschee et al. 2015) contain millions of facts, they remain incomplete due to the evolving nature of knowledge (Wang et al. 2024b). This has made Temporal Knowledge Graph Completion (TKGC) become a key task, which aims to infer missing facts, thereby enhancing the completeness and reliability of TKGs. Traditional TKGC works mainly fall into two categories: temporal logic rule-based model (e.g., TLogic (Liu et al. 2022)) and deep learning-based model (e.g., RE-GCN (Li et al. 2021)). Despite their effectiveness, these methods have several limitations. First, static temporal logic rules struggle to model complex temporal patterns. Moreover, GNNs are susceptible to noise caused by errors in automatic information extraction system, resulting in suboptimal graph learning (Ding et al. 2023). Finally, they rely on high-quality data, limiting their knowledge in open-world scenarios (Chang et al. 2025).

Given these limitations, the reasoning and generalization ability of Large Language Models (LLMs) (Ouyang et al. 2022; Touvron et al. 2023) offer promising directions for TKGC and have been explored by recent studies, which generally fall into two categories: text-based and embedding-based approaches. Text-based methods leverage LLMs' generative and reasoning abilities by designing a variety of temporal retrieval strategies. For example, COH (Luo et al. 2024) uses enriched historical cues to identify key temporal nodes; GenTKG (Liao et al. 2024) constructs a rule base guided by temporal logic; LLM-DA (Wang et al. 2024c) extracts such rules via constrained Markov random walks. However, TKGs often contain complex dependencies and extensive historical data, due to token length limit, incorporating full context is impractical. Moreover, the structural-textual gap may cause incomplete retrieval, thus omitting crucial temporal and structural signals. To overcome these issues, embedding-based approaches (Zhang et al. 2024a,b) project graph-encoded structures into the language token space of LLMs, enabling implicit structural understanding. Compared to text-based methods, they offer richer and more complete context modeling. Despite these advantages, current embedding-based methods still suffer from limitations:

- **Insufficient modeling of temporal logic:** Existing methods have made progress in static KGs reasoning, but overlook temporal patterns in TKGs. Without access to historical graphs and context, LLMs struggle to perform accurate temporal reasoning.
- **Insufficient generalization and suboptimal graph representation:** Existing methods rely on ideal graph models to obtain graph embeddings as idealized structural

*Corresponding author.

representations. However, obtaining ideal graph models requires high-quality labeled data, which limits generalization in open-world domains. Meanwhile, when the graph models are suboptimal due to the noise and long-tail issues, the reasoning ability of LLMs are limited.

- **Insufficient temporal cross-modal alignment between graph and language:** Existing methods emphasize graph-language alignment but fail to incorporate temporal dimension that are essential for TKGC reasoning. Without temporal constraints, LLMs may incorrectly align similar structures with language across different timestamps, resulting in temporal reasoning errors.

To address these challenges, we propose a novel framework, TGCA-LLM, which integrates LLMs and TKGs through a carefully designed Graph-Text-Temporal contrastive alignment module and a two-stage instruction tuning strategy. First, to capture graph structures and temporal patterns, we use RGCN (Schlichtkrull et al. 2018) to extract relational structures at each timestamp, and employ Gated Recurrent Unit (GRU) to model temporal evolution, thus extracting historical graph embeddings to retain the temporal information of fact evolution. Second, to obtain a generalizable and robust temporal graph encoder without high-quality labeled data, we integrate time-aware Graph-Text contrastive learning into graph learning to obtain initially aligned graph embeddings. It utilizes contrastive learning with temporal dimension to align graph structures with corresponding natural language. Finally, to enhance temporal cross-modal alignment, we employ a two-stage instruction tuning strategy to guide LLMs. Specifically, we first introduce a lightweight graph projector to map pretrained graph embeddings into the LLMs token space. Then we generate self-supervised alignment signals by training LLMs to match graph tokens with corresponding text descriptions under time constraints, enhancing LLMs' comprehension of graph structures and temporal patterns. In the second stage, we concatenate graph tokens chronologically and augment them with relevant historical context, forming structured inputs for TKGC-specific instruction tuning. Extensive experiments show that our method outperforms a range of state-of-the-art (SOTA) baselines. In summary, our method makes the following contributions:

- We propose TGCA-LLM, a novel framework that integrates time-aware structured graph knowledge into LLMs without relying on high-quality data, enhancing their reasoning capabilities for TKGC tasks.
- We introduce a carefully designed Graph-Text-Temporal contrastive alignment module and a two-stage instruction tuning strategy, which respectively alleviate the three limitations of existing LLM-based TKGC methods.
- We conducted extensive experiments on three widely used real-world datasets. The experimental results validated the effectiveness of TGCA-LLM.

Related Work

Traditional non-LLM Methods Traditional methods are broadly divided into rule-based and deep learning-based approaches. Rule-based methods extract temporal logic patterns for symbolic reasoning, such as using random walks (Liu et al. 2022), pruning-based rule selection (Bai et al. 2023), and differentiable induction (Xiong et al. 2023). While interpretable, these methods struggle with complex, evolving dynamics. Deep learning-based methods learn latent temporal representations for prediction. Extensions of TransE (Jiang et al. 2016) like TTransE (Bordes et al. 2013) and TA-TransE (García-Durán, Dumančić, and Niepert 2018) incorporate time into embeddings, while RE-NET (Jin et al. 2020a) and ChronoR (Sun et al. 2019; Sadeghian et al. 2021) leverage RNNs or rotations for temporal modeling. Graph-based models such as RGCN (Schlichtkrull et al. 2018), RE-GCN (Li et al. 2021), and ConvTKG (He, Zhu, and Bai 2024) integrate GNNs with temporal encoders to model graph structure and time jointly. However, their reliance on high-quality labels and sensitivity to noisy or sparse data limits robustness and generalization.

LLMs-based Methods Recent works apply LLMs to TKGC by leveraging their contextual reasoning abilities. These approaches fall into two main categories: text-based and embedding-based. Text-based methods convert graph data into natural language and rely on LLMs for inference. Techniques include historical fact retrieval (Luo et al. 2024), temporal logic-guided prompt construction (Liao et al. 2024), fact list formatting (Lee et al. 2023b), and rules generation (Wang et al. 2024c; Chen et al. 2025). Though effective, these methods suffer from input length limits and loss of structural fidelity. Embedding-based methods align graph representations with LLM token embeddings for cross-modal reasoning. Representative approaches include feature projection (Zhang et al. 2024a), translation modules (Zhang et al. 2024b), and adapter-based integration (Wang et al. 2024a). While these methods improving structural alignment, most overlook temporal patterns. TGL-LLM (Chang et al. 2025) addresses this by modeling temporal embeddings and simulating time in prompts. Nevertheless, these methods rely on high-quality pre-trained graph models, hindering generalization and alignment quality.

Methodology

In this section, we introduce TGCA-LLM, a novel framework designed to enhance TKGC performance by integrating structural and temporal knowledge from TKGs into LLMs through dynamic multimodal alignment. As shown in Figure 1, TGCA-LLM contains two core modules: (1) a Graph-Text-Temporal contrastive alignment module that captures temporal patterns and aligns structural embeddings with corresponding historical textual descriptions. (2) a two-stage instruction tuning strategy that guides LLMs to internalize time-aware graph semantics for reasoning.

Graph-Text-Temporal Contrastive Alignment

This module contains two main components: (1) Temporal Graph Encoding, which introduces how to obtain historical graph embeddings. (2) Time-Aware Graph-Text Contrastive Alignment, which demonstrates our temporal graph contrastive learning process.

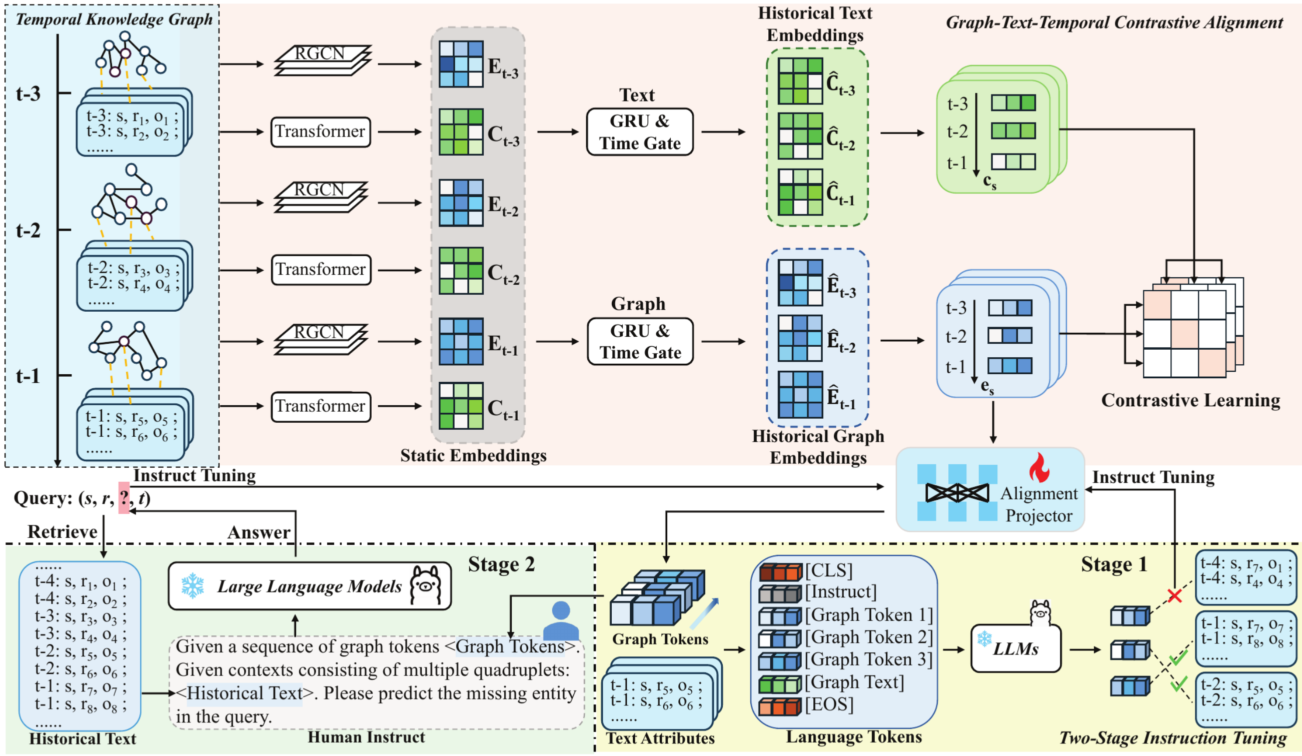

Figure 1: Overview of the TGCA-LLM Framework. The framework consists of two key modules: (1) a Graph-Text-Temporal contrastive alignment module (top), which incorporates temporal patterns and textual semantic into the graph structure; (2) a two-stage instruction tuning module (bottom), designed to enhance the LLMs' temporal reasoning and structural understanding.

Temporal Graph Encoding Following previous studies (Li et al. 2021), we employ RGCN to encode structural dependencies among entities at each timestamp t. Given a snapshot $\mathcal{G}_t$ at timestamp t, we update entity embeddings through message-passing framework. Formally, at l-th layer, the embedding of entity o is defined as:

$$h_o^l = f\left(\frac{1}{|\mathcal{G}_t|}\sum_{(s,r,o)\in\mathcal{G}_t} W_1^{l-1}(h_s^{l-1}+r) + W_2^{l-1}h_o^{l-1}\right) \quad (1)$$

Where $f(\cdot)$ denotes the activation function RReLU (Xu et al. 2015), W_1^{l-1} and $W_2^{l-1} \in \mathbf{R}^d$ are the learnable parameters at layer $l-1$. The variable d represents the dimension of the embedding. Moreover, h^{l-1} and r correspond to the embeddings of the entity, and the relation r, respectively. We then aggregate all L layers to obtain the final representation $e_o = \sum_{l=0}^{L} h_o^l$, and construct the entity embedding matrix $E_t \in \mathbf{R}^{|\mathcal{E}|\times d}$ for time t, where $|\mathcal{E}|$ denotes the total number of entities in TKGs. To capture temporal patterns, we collect embeddings from the recent T steps, denoted as historical graph embeddings $E = \{E_{t-T}, ..., E_t\}$.

To model temporal evolution patterns, we employ GRU to capture sequential dependencies:

$$\hat{E}_t = \mathrm{GRU}(E_t, E_{t-1}) \quad (2)$$

Furthermore, to mitigate over-smoothing and gradient vanishing (Li et al. 2019) while enhancing temporal sensitivity of the model, we introduce a time gated update:

$$\begin{aligned} \dot{E}_t &= U_t \otimes \hat{E}_t^L + (1-U_t)\otimes \hat{E}_{t-1} \\ U_t &= \sigma(W_3\hat{E}_{t-1}+b) \end{aligned} \quad (3)$$

Where $\otimes$ denotes the element-wise product, $\hat{E}_t^L$ is the output of the final RGCN layer, U_t controls the temporal blending of current and previous representations, and $\sigma(\cdot)$ denotes the sigmoid activation function.

Time-Aware Graph-Text Contrastive Alignment To initially bridge the gap between structured temporal graphs and LLMs' language space, we design a temporal contrastive alignment paradigm that integrates textual semantics, graph structures, and temporal evolution. Inspired by (Radford et al. 2021; Wen and Fang 2023; Tang et al. 2024), we apply contrastive learning across multiple levels of granularity.

Given a TKG $\mathcal{G}_T = \{\mathcal{G}_{t-T}, \ldots, \mathcal{G}_t\}$ over T time steps, we collect the corresponding raw textual context $C_t = \{c_t^1, \ldots, c_t^K\}$ for each entity from related quadruples at each timestamp t, where $|K|$ is the number of nodes in $\mathcal{G}_t$. The graph and text encoders $f_G(\cdot)$ and $f_C(\cdot)$ then produce graph and text representations:

$$E_t = f_G(\mathcal{G}_t), \quad C_t' = f_C(C_t) \quad (4)$$

In this paper, we use a Transformer with time-step embedding as the text encoder. Similar to the graph encoding, we collect text embeddings from the recent T steps, denoted as historical text embeddings $C = \{C'_{t-T}, \ldots, C'_t\}$. Then, we also use a GRU to capture the temporal pattern of each text embeddings, and normalize the graph and text embeddings to ensure consistency of scale during contrastive learning:

$$\begin{aligned} \hat{E}_t &= \mathrm{GRU}(E_t, E_{t-1}), \quad \hat{C}_t = \mathrm{GRU}(C'_t, C'_{t-1}) \\ \tilde{E}_t &= \mathrm{norm}(\dot{E}_t), \quad \tilde{C}_t = \mathrm{norm}(\hat{C}_t) \end{aligned} \tag{5}$$

Building on the previous steps, we derive normalized temporal graph embeddings and corresponding text embeddings $\tilde{E}, \tilde{C} \in \mathbf{R}^{B \times T \times d}$, where B denotes the number of input entities, and d denotes the embedding dimension. To optimize alignment in the time dimension, we used a multi-perspective contrastive loss:

$$\begin{aligned} \mathcal{L} &= L_{\text{time}} + L_{\text{cm}} + L_{\text{ent}} + L_{\text{win}} \\ &= \sum_i \frac{1}{2} \lambda_i (\mathtt{CE}(\Gamma_i, y_i) + \mathtt{CE}(\Gamma_i^\top, y_i)) \\ \Gamma_i &= s \cdot E_i C_i^\top, \quad i \in \{\text{time}, \text{cm}, \text{ent}, \text{win}\} \end{aligned} \tag{6}$$

Where Γ_i is a pairwise similarity matrix between graph and text representations, scaled by a learnable temperature $s = \exp(\tau)$. Each loss component corresponds to a distinct alignment view: temporal (L_{time}), cross-modal (L_{cm}), entity-level (L_{ent}), and window-based event-level (L_{win}). To construct training pairs, historical graph and textual embeddings $\tilde{E}, \tilde{C}$ are reshaped into multiple views. Specifically, temporal embeddings $E_{time}, C_{time} \in \mathbf{R}^{T \times S}$, with $S = B \times d$, enable fine-grained time alignment, while $E_{cm}, C_{cm} \in \mathbf{R}^{N \times d}$, with $N = B \times T$, support global cross-modal alignment. Entity-level representations $E_{ent}, C_{ent} = \frac{1}{T} \sum_{t=1}^{T} \tilde{E}_t, \tilde{C}_t$. For event-level contrast, sliding windows of width W are applied to temporal sequences, resulting in $E_{\text{win}}, C_{\text{win}} \in \mathbf{R}^{M \times d}$, where $M = B \times (T - W + 1)$.

Each perspective uses a label vector y_i indicating the ground truth alignment index (e.g., $y_{time} = (0, 1, \ldots, T-1)^\top$). This multi-perspective contrastive loss guides the graph encoder to integrate temporal and textual semantic signals more effectively, improving cross-modal fusion with LLMs for downstream TKGC reasoning.

Two-Stage Instruction Tuning

Temporal Graph Projector To enable LLMs to interpret temporal graph structures, we map graph embeddings into the LLM token space via a lightweight two-layer perceptron projector. Given the historical embeddings of a subject entity $e_s \in \mathbf{R}^{T \times d}$, the projector f_p outputs temporally ordered graph tokens:

$$Z_{\mathcal{G}} = f_p(e_s \mid \theta_p) \tag{7}$$

where $Z_{\mathcal{G}} \in \mathbf{R}^{T \times D}$ represents the projected graph tokens and θ_p denotes the learnable parameters of f_p. These tokens replace the placeholder $< \texttt{GraphTokens} >$ in the prompt with a structured sequence: $\{< \texttt{graph\_begin} >, < \texttt{graph\_token} >_1, \ldots, < \texttt{graph\_token} >_T, < \texttt{graph\_end} >\}$. This projection bridges the modality gap,

Prompt for Stage 1

You are a large language and temporal-graph-structural assistant. Given a sequence of graph tokens <**GraphTokens**> that constitute the temporal subgraph of a target entity ***s***. Here is a list of facts: $\{\mathcal{F}_s^{t-T}, \ldots, \mathcal{F}_s^{t-1}, \mathcal{F}_s^{t}\}$.
Please reorder the list of facts according to the order of graph tokens. **(a)**

Prompt for Stage 2

You are a large language and temporal-graph-structural assistant. Given a subgraph of a temporal knowledge graph: <**GraphTokens**>. The historical context are as follows: $t-T$: $[s, r_i, o_j]$, ..., $t-1$: $[s, r_1, o_1]$, Query: t: $[s, r, ?]$.
Please predict missing entities based on graph and historical context. **(b)**

Figure 2: Two simple examples of the two stages. (a) Stage 1 example. (b) Stage 2 example.

injecting temporal graph semantics into language prompts and enabling LLMs to reason over structured history.

Temporally-Constrainted Graph Matching Instruction Tuning In the first stage, we adopt a self-supervised instruction tuning approach inspired by (Tang et al. 2024), which leverages unlabeled structural and temporal signals from TKGs to construct instruction-response pairs. Specifically, we formulate a graph matching task where the LLMs must match the graph tokens with their corresponding natural language descriptions at the correct timestamps and reorder the texts chronologically. This design enables the model to learn both structural semantics and temporal ordering, enhancing its understanding of TKGs.

Prompt Design: Each prompt consists of three components: (1) a sequence of projected graph tokens representing temporal subgraphs of a target entity, (2) human instructions indicating the alignment task, and (3) a list of unordered natural language fact descriptions. For a given entity s and time window $[t-T, t]$, we construct subgraphs from its fact history $\mathcal{F}_s = \{\mathcal{F}_s^{t-T}, \ldots, \mathcal{F}_s^{t}\}$, encode them using the graph encoder, and insert them into the prompt via a placeholder token $< \texttt{GraphTokens} >$. The textual input includes fact-level descriptions (e.g., "Entity A *Consult* Entity B in t") corresponding to the same time window. To prevent shortcut learning based on timestamps alone, we randomly shuffle the graph token sequence and require the LLM to restore the correct alignment based on temporal and structural semantics. This matching task forces the model to reason over the implicit time order and content of graph tokens, grounding LLMs in the temporal logic of the TKG. Figure 2(a) shows a sample prompt example.

Training Strategy: As described in the previous section, we use the projector f_p to map graph embeddings into the LLMs' language token space. During training, the parameters of the graph encoder and LLM are frozen, and only the projector parameters θ_p are updated. The goal is to learn an effective mapping from structured graph embeddings to temporally ordered graph tokens aligned with entity-related text. Since the temporal graph matching task is unsupervised, we can exploit a large volume of unlabeled entity data across different time intervals to improve the projector's generalization and performance. Let $Z_{\mathcal{G}} = f_p(e_s)$ denote the pro-

Type	Model	ICEWS14				ICEWS05-15				ICEWS18			
		MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Non-LLMs	DistMult (2015)	0.203	0.061	0.276	0.466	0.199	0.056	0.272	0.473	0.139	0.056	0.152	0.313
	TTransE (2016)	0.129	0.031	0.297	0.337	0.165	0.055	0.208	0.393	0.084	0.019	0.090	0.224
	TA-DistMult (2018)	0.262	0.168	0.297	0.452	0.275	0.176	0.315	0.473	0.164	0.086	0.181	0.325
	RGCN (2018)	0.280	0.194	0.320	0.448	0.271	0.188	0.304	0.432	0.151	0.081	0.165	0.290
	RE-NET (2020)	0.369	0.268	0.395	0.548	0.433	0.335	0.478	0.631	0.288	0.191	0.324	0.475
	RE-GCN (2021)	0.404	0.307	0.448	0.592	0.480	0.373	0.539	0.683	0.306	0.210	0.343	0.488
	HiSMatch (2022)	0.464	0.359	0.516	0.668	0.529	0.420	0.591	0.733	0.340	0.239	0.379	0.538
LLMs	PPT (2023)	0.384	0.289	0.425	0.570	0.389	0.286	0.434	0.586	0.266	0.169	0.306	0.454
	DeepSeek-V3 (2024)	0.454	0.364	0.479	0.657	0.487	0.382	0.576	0.650	0.345	0.273	0.352	0.414
	GPT-4o-mini (2024)	0.459	0.368	0.484	0.664	0.474	0.372	0.560	0.632	0.336	0.266	0.343	0.403
	COH (2024)	0.424	0.338	0.462	0.587	0.469	0.370	0.531	0.699	0.313	0.219	0.361	0.520
	CenTKG (2024)	0.438	0.349	0.473	0.619	0.461	0.360	0.525	0.687	0.309	0.215	0.366	0.496
	LLM-DA (2024)	0.471	0.369	0.526	$\underline{0.671}$	0.521	0.416	0.586	0.728	0.357	0.255	0.403	$\underline{0.570}$
	KoPA (2024)	0.439	0.353	0.472	0.602	0.478	0.376	0.540	0.711	0.359	0.279	$\underline{0.419}$	0.564
	LLM-DR (2025)	$\underline{0.505}$	$\underline{0.406}$	$\underline{0.558}$	0.670	$\underline{0.589}$	$\underline{0.505}$	$\underline{0.648}$	**0.753**	$\underline{0.382}$	$\underline{0.304}$	0.409	0.556
	TGCA-LLM (ours)	**0.549**	**0.482**	**0.591**	**0.693**	**0.650**	**0.602**	**0.683**	$\underline{0.752}$	**0.476**	**0.433**	**0.496**	**0.591**

Table 1: Comparison results between TGCA-LLM and non-LLM methods as well as LLM-based methods. Bold indicates the best performance, and underlined indicates the second-best performance.

jected graph tokens, and $Z_{\mathcal{I}} = tokenizer(instruction)$ represent the tokenized instructions. For an instruction sequence of length N, the probability of generating the expected output $Z_{\mathcal{O}}$ can be defined as:

$$p(Z_{\mathcal{O}}|Z_{\mathcal{G}}, Z_{\mathcal{I}}) = p_\theta(Z_{\mathcal{O}}|\mathcal{C}) = \prod_{t=1}^{N} p_\theta(z_t|\mathcal{C}_{<t}) \quad (8)$$

where $\mathcal{C} = [Z_{\mathcal{G}}; Z_{\mathcal{I}}]$ is the concatenated context, $\mathcal{C}_{<t} = [Z_{\mathcal{G}}; Z_{\mathcal{I}}; Z_{\mathcal{O},<t}]$ includes all tokens preceding timestep t, i.e., graph tokens, instruction tokens, and previously generated outputs. This conditional generation objective encourages the model to ground textual reasoning in temporally structured graph inputs.

TKGC Instruction Tuning In the second stage, we conduct task-specific instruction tuning for TKGC to further enhance LLMs' temporal reasoning and prediction capabilities. By incorporating graph tokens and TKGC-oriented instructions, the model learns to generate answers grounded in both structural and temporal context.

Prompt Design: We follow the same prompt format as in the first stage, consisting of three parts: (1) temporal graph embeddings, (2) human instruction, and (3) expected response. Given a query $(s, r, ?, t)$, we retrieve the historical graph embeddings of the subject entity s over the past T timestamps, project them into the language token space via the trained projector f_p, and insert them at the $<$ `GraphTokens` $>$ position. The instruction includes the indicator token and a natural language list of historical facts involving s, where each fact (s, r, o, t) is textualized as "*timestamp*: [*subject*, *relation*, *object*]" (Luo et al. 2024). This design supports reasoning about the missing object entity using both structural and temporal cues. Figure 2(b) shows a sample example.

Training Strategy: We initialize the projector f_p using weights learned from the first stage and freeze both the graph encoder and LLMs. Only the projector parameters are updated, allowing f_p to specialize in mapping graph embeddings to token representations optimized for the TKGC task.

Experiments

Our experiment mainly answers the following three questions and provides an intuitive case study:

- **RQ1:** How does TGCA-LLM perform compared to traditional non-LLM methods and LLM-based baselines?
- **RQ2:** What is the impact of temporal contrastive alignment and temporally-constrained instruction tuning?
- **RQ3:** How does TGCA-LLM's performance vary under different historical text and graph lengths?

Experimental Settings

Datasets We use the ICEWS14, ICEWS05-15 (García-Durán, Dumančić, and Niepert 2018), and ICEWS18 (Jin et al. 2020b) datasets for evaluation. ICEWS14, ICEWS05-15, and ICEWS18 are subsets of the Integrated Crisis Early Warning System (ICEWS), which represent TKGs of international political events and social dynamics.

Evaluation Metrics To evaluate model performance, we adopt widely used metrics in our experiments: Mean Reciprocal Rank (MRR) and Hits@N ($N = 1, 3, 10$), where higher values indicate better performance. Without loss of generality, all results are reported under the raw setting.

Baselines We compare our method with two categories of baselines: non-LLMs methods and LLMs-based methods. **Non-LLMs:** Traditional embedding-based models, including DistMult (Yang et al. 2015), TTransE (Jiang et al. 2016), TA-DistMult (García-Durán, Dumančić, and Niepert 2018), RGCN (Schlichtkrull et al. 2018), RE-NET (Jin et al. 2020a), RE-GCN (Li et al. 2021), and HiSMatch (Li et al. 2022). **LLMs:** Zero-shot methods prompt LLMs with textualized facts (DeepSeek-V3 (DeepSeek-AI 2024), GPT-4o-mini (OpenAI 2024)); Fine-tuning methods train LLMs with

Models	ICEWS14		ICEWS05-15		ICEWS18	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
w/o-TO	0.347	0.297	0.461	0.377	0.331	0.271
w/o-SG	0.442	0.363	0.483	0.412	0.360	0.282
w/o-CT	0.486	0.403	0.544	0.503	0.402	0.354
TGCA-LLM	**0.549**	**0.482**	**0.650**	**0.602**	**0.476**	**0.433**

Table 2: The impact of different graph representation forms.

Models	ICEWS14		ICEWS05-15		ICEWS18	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
w/o-DT	0.501	0.424	0.611	0.568	0.427	0.378
w/o-FO	0.493	0.418	0.596	0.554	0.419	0.371
TGCA-LLM	**0.549**	**0.482**	**0.650**	**0.602**	**0.476**	**0.433**

Table 3: The impact of two-stage instruction tuning forms.

textual or graph inputs, including PPT (Xu et al. 2023), COH (Luo et al. 2024), CenTKG (Liao et al. 2024), LLM-DA (Wang et al. 2024c), KoPA (Zhang et al. 2024b), and LLM-DR (Chen et al. 2025).

Performance Comparison (RQ1)

We compare TGCA-LLM with state-of-the-art TKGC methods, including both traditional embedding-based and LLM-based approaches. Table 1 presents the experimental results.

First, as expected, TGCA-LLM outperforms all baselines across three datasets, demonstrating the effectiveness of our method. Furthermore, we conducted ANOVA significance tests and find that all p-values are below 0.01, confirming the robustness of these improvements. Second, we find that some LLM-based TKGC methods do not always surpass traditional methods (e.g., HiSMatch (Li et al. 2022)), indicating that the rules generated by LLM are too broad and may be difficult to adapt precisely to specific data. In contrast, TGCA-LLM effectively adapts LLMs to TKG tasks by explicitly injecting structural and temporal patterns. Moreover, despite using the smaller Llama-2-7b (Touvron et al. 2023), TGCA-LLM still outperforms two large-scale commercial LLMs (e.g., DeepSeek-V3 with 671B parameters). This emphasizes the benefit of incorporating structured graph signals and tailored instruction tuning, rather than relying solely on massive model scale. Finally, KoPA (Zhang et al. 2024b) is an important baseline as it also integrates graph structure into LLMs. Nevertheless, TGCA-LLM shows significant improvements. This is because KoPA is designed for static KGs and lacks temporal modeling, confirming the importance of temporal alignment and instruction-aware adaptation in LLMs' reasoning.

Ablation Study (RQ2)

Impact of Different Graph Representation Forms To investigate how different graph representations affect LLM reasoning, we compare three TGCA-LLM variants: (1) w/o-TO, which uses only textual descriptions without structural input. (2) w/o-SG, which introduces static graph structure. (3) w/o-CT, which adopts ConvTransE-based graph representations from RE-GCN. Results are shown in Table 2.

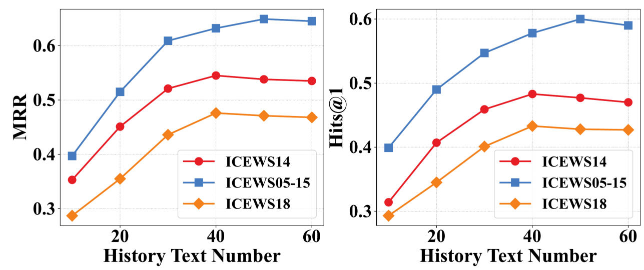

Figure 3: The performance comparison of different historical text quantities across three datasets.

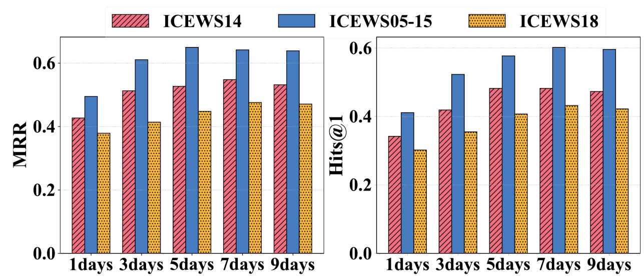

Figure 4: The performance comparison of different historical graph length across three datasets.

As clearly shown in Table 2, using only plain text (w/o-TO) yields the worst performance, indicating that LLMs struggle to capture structural dependencies from text alone. Secondly, adding static graphs (w/o-SG) helps but still underperforms due to the lack of temporal dynamics. Furthermore, deep representations from RE-GCN (w/o-CT) do not offer further gains, likely because LLMs struggles to effectively align deeper-level graph representations with language tokens across modalities. In contrast, TGCA-LLM, with explicit temporal modeling and Graph-Text alignment, achieves the best results, confirming that time-aware and language-aligned graph signals are crucial for reasoning.

Impact of the Two-Stage Tuning To evaluate our two-stage tuning strategy, we compare TGCA-LLM with two variants: w/o-DT, which skips the first stage and directly tunes on the TKGC task, and w/o-FO, which uses fixed graph token order without shuffling.

Results in Table 3 show that TGCA-LLM outperforms both variants, confirming the importance of two-stage tuning. Notably, the drop in performance for w/o-FO suggests that without token shuffling, LLMs may exploit superficial timestamp cues in the text rather than learning intrinsic temporal patterns from graph structure. These results demonstrate that temporally-constrained graph matching task is crucial for teaching LLMs to reason over structured and temporal data effectively.

Exploration on History Length (RQ3)

We further explore how historical context affects LLMs' reasoning by varying the amount of fact text and graph length.

Query	(Philip Hammond, Make statement, ?, 236)
Ground-Truth	Iraq
Instruction to TGCA-LLM	Given a subgraph of a temporal knowledge graph $< \texttt{GraphTokens} >$. The subgraph consists of facts in the form of ”timestamp: [subject, relation, object]” that occured on entity ”Philip Hammond”. You must be able to correctly predict the next ”object” from the query in the form of ”timestamp: [subject, relation,” in the end based on the graph tokens and the historical context. You must output the top 10 ”object” that you think may be the answer (one per line) from the historical context and graph tokens to the given query without explanations. Note that answers with high probability should be listed first. The historical context are as follows:
Instruction to Deepseek-V3 and GPT-4o-mini	Given a sequence of facts in the form of ”timestamp: [subject, relation, object]” that occured on entity ”Philip Hammond”. You must be able to correctly predict the next ”object” from the query in the form of ”timestamp: [subject, relation,” in the end based on the historical context. You must output the top 10 ”object” that you think may be the answer (one per line) from the historical context to the given query without explanations. Note that answers with high probability should be listed first. The historical context are as follows:
Historical context	205: [Philip Hammond, Make a visit, Abdel Fattah Al-Sisi]; 212: [Philip Hammond, Praise or endorse, John Kerry]; 220: [Philip Hammond, Consult, Sergey Viktorovich Lavrov]; 222: [Philip Hammond, Discuss by telephone, Sergey Viktorovich Lavrov]; 226: [Philip Hammond, reverse Praise or endorse, Nuri al-Maliki]; 226: [Philip Hammond, Make a visit, Iraq]; 226: [Philip Hammond, reverse Praise or endorse, Iraq]; 235: [Philip Hammond, Make statement, Militant (Iraq)]; 235: [Philip Hammond, Make statement, Royal Air Force]
TGCA-LLM response	1. **Iraq**; 2. Militant (Iraq); 3. Nuri al-Maliki; 4. Royal Air Force; 5. Abdel Fattah Al-Sisi.
Deepseek-V3 response	1. Militant (Iraq); 2. Royal Air Force; 3. Abdel Fattah Al-Sisi; 4. **Iraq**; 5. Sergey Viktorovich Lavrov.
GPT-4o-mini response	1. Militant (Iraq); 2. Royal Air Force; 3. **Iraq**; 4. John Kerry; 5. Nuri al-Maliki.

Table 4: A case study comparing the prediction results of our TGCA-LLM with ChatGPT and DeepSeek. Includes instructions with graph tokens and plain text instructions. The bold text is the correct answer.

Impact of the Auxiliary Historical Text While LLMs are now capable of directly interpreting graph structure data, relying solely on such representations for prediction remains insufficient. To enrich model context, we add auxiliary historical textual descriptions of the query subject entity s to the prompt. We hypothesize that these supplementary texts significantly contribute to the model's reasoning capabilities. To validate this hypothesis, we vary the number of supporting historical facts (i.e., 10, 20, 30, 40, 50, and 60 quadruples) while keeping other settings fixed. The results are shown in Figure 3.

Figure 3 show that model performance improves as the number increases up to 30, then plateaus or slightly declines. This phenomenon indicates that providing moderate historical context enhances LLMs reasoning. In addition, we also observe that supplying too much historical text leads to performance degradation. This decline is likely due to excessive input introducing noise that hinders LLMs reasoning.

Impact of the Historical Graph Length The temporal scope of historical graph data is critical for enabling LLMs to perform accurate reasoning. To examine its influence, we conduct experiments by varying the historical length of graph embeddings across five fixed values (1, 3, 5, 7, and 9), while maintaining all other settings constant. The corresponding results are presented in Figure 4.

As illustrated in Figure 4, model performance improves across all datasets when the historical graph length is less than 5, then stabilizes or drops slightly. Notably, history lengths of 5 and 7 yield the best results across the datasets. This phenomenon indicates that longer history windows may introduce redundant or less relevant structural patterns, reducing reasoning effectiveness. The above results confirm that both textual and structural histories are helpful, but overly long contexts can be counterproductive.

Case Study

To intuitively compare TGCA-LLM with traditional LLM-based methods, we evaluate TGCA-LLM, DeepSeek-V3, and GPT-4o-mini on a sample from ICEWS14 using identical historical contexts and similar instructions.

Results in Table 4 clearly shows the output results of the three models. TGCA-LLM ranks the correct answer first, while DeepSeek-V3 and GPT-4o-mini place it fourth and third, respectively. This highlights that even large-scale LLMs (e.g., DeepSeek-V3 with 671B parameters) struggle with fine-grained reasoning based solely on textual context. This challenge is especially evident when dealing with historical information that contains similar relations. In contrast, TGCA-LLM, despite using a smaller Llama-2-7b, achieves accurate predictions by incorporating structural signals with time patterns through projected graph tokens and leveraging temporal reasoning via two-stage tuning.

Conclusion

In this paper, we propose TGCA-LLM, a novel framework for TKGC task, which integrates two core components: Graph-Text-Temporal Contrastive Alignment and Two-Stage Instruction Tuning. The former injects temporal and textual context into graph representations, while the latter guides LLMs to learn temporal logic and structural patterns through staged instruction tuning. TGCA-LLM effectively addresses challenges in embedding-based LLM methods: limited temporal modeling, reliance on high-quality data, and weak cross-modal alignment. Experimental results confirm its superior performance over existing baselines, highlighting its effectiveness and competitive performance. In future work, we will explore more flexible graph adapters beyond MLPs and extend TGCA-LLM to broader downstream tasks with improved engineering practicality.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundations of China under Grant (62372060).

References

Bai, L.; Yu, W.; Chai, D.; Zhao, W.; and Chen, M. 2023. Temporal knowledge graphs reasoning with iterative guidance by temporal logical rules. *Information Sciences*, 621: 22–35.

Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the Twenty-Seventh International Conference on Neural Information Processing Systems*, 2787–2795. Red Hook, NY, USA: Curran Associates Inc.

Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; and Ward, M. 2015. ICEWS Coded Event Data.

Chang, H.; Wu, J.; Tao, Z.; Ma, Y.; Huang, X.; and Chua, T.-S. 2025. Integrate Temporal Graph Learning into LLM-based Temporal Knowledge Graph Model. arXiv:2501.11911.

Chen, K.; Song, X.; Wang, Y.; Gao, L.; Li, A.; Zhao, X.; Zhou, B.; and Xie, Y. 2025. LLM-DR: A Novel LLM-Aided Diffusion Model for Rule Generation on Temporal Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(11): 11481–11489.

DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437.

Ding, Z.; Li, Z.; Qi, R.; Wu, J.; He, B.; Ma, Y.; Meng, Z.; Chen, S.; Liao, R.; Han, Z.; and Tresp, V. 2023. ForecastTKGQuestions: A Benchmark for Temporal Question Answering and Forecasting over Temporal Knowledge Graphs. In *Proceedings of the 22nd International Semantic Web Conference*, 541–560. Berlin, Heidelberg: Springer-Verlag.

Feng, C.; Zhang, X.; and Fei, Z. 2023. Knowledge Solver: Teaching LLMs to Search for Domain Knowledge from Knowledge Graphs. arXiv:2309.03118.

García-Durán, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the Twenty-Third Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4816–4821. Brussels, Belgium: Association for Computational Linguistics.

Han, Z.; Chen, P.; Ma, Y.; and Tresp, V. 2021. Explainable Subgraph Reasoning for Forecasting on Temporal Knowledge Graphs. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*.

He, M.; Zhu, L.; and Bai, L. 2024. ConvTKG: A query-aware convolutional neural network-based embedding model for temporal knowledge graph completion. *Neurocomputing*, 588: 127680.

Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Li, S.; Chang, B.; and Sui, Z. 2016. Encoding Temporal Information for Time-Aware Link Prediction. In *Proceedings of the Twenty-First Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2350–2354. Austin, Texas: Association for Computational Linguistics.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2020a. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the Twenty-Fifth Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6669–6683. Online: Association for Computational Linguistics.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2020b. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the Twenty-Fifth Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6669–6683. Online: Association for Computational Linguistics.

Lee, D.-H.; Ahrabian, K.; Jin, W.; Morstatter, F.; and Pujara, J. 2023a. Temporal Knowledge Graph Forecasting Without Knowledge Using In-Context Learning. In *Proceedings of the Twenty-Eighth Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 544–557. Singapore: Association for Computational Linguistics.

Lee, D.-H.; Ahrabian, K.; Jin, W.; Morstatter, F.; and Pujara, J. 2023b. Temporal Knowledge Graph Forecasting Without Knowledge Using In-Context Learning. In *Proceedings of the Twenty-Eighth Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 544–557. Singapore: Association for Computational Linguistics.

Li, G.; Müller, M.; Thabet, A.; and Ghanem, B. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs? In *Proceedings of the Seventeenth IEEE/CVF International Conference on Computer Vision (ICCV)*, 9266–9275. IEEE. KAUST Repository Item: Exported on 2020-10-01.

Li, Y.; Sun, S.; and Zhao, J. 2022. TiRGN: Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2152–2158. IJCAI Organization.

Li, Z.; Hou, Z.; Guan, S.; Jin, X.; Peng, W.; Bai, L.; Lyu, Y.; Li, W.; Guo, J.; and Cheng, X. 2022. HiSMatch: Historical Structure Matching based Temporal Knowledge Graph Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 7328–7338. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; and Cheng, X. 2021. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning. In *Proceedings of the Forty-Fourth International ACM SIGIR Conference on Research and Development in Information Retrieval*, 408–417. New York, NY, USA: Association for Computing Machinery.

Liao, R.; Jia, X.; Li, Y.; Ma, Y.; and Tresp, V. 2024. GenTKG: Generative Forecasting on Temporal Knowledge Graph with Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 4303–4317. Mexico City, Mexico: Association for Computational Linguistics.

Liu, P.; Qian, B.; Sun, Q.; and Zhao, L. 2023. Prompt-WNQA: A prompt-based complex question answering for wireless network over knowledge graph. *Computer Networks*, 236: 110014.

Liu, Y.; Ma, Y.; Hildebrandt, M.; Joblin, M.; and Tresp, V. 2022. TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4): 4120–4127.

Luo, R.; Gu, T.; Li, H.; Li, J.; Lin, Z.; Li, J.; and Yang, Y. 2024. Chain of History: Learning and Forecasting with LLMs for Temporal Knowledge Graph Completion. arXiv:2401.06072.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the Thirty-Sixth International Conference on Neural Information Processing Systems*, 27730–27744. Curran Associates, Inc.

Qin, Y.; Gao, C.; Wei, S.; Wang, Y.; Jin, D.; Yuan, J.; Zhang, L.; Li, D.; Hao, J.; and Li, Y. 2023. Learning from hierarchical structure of knowledge graph for recommendation. *ACM Transactions on Information Systems*, 42(1): 1–24.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the Thirty-eighth International Conference on Machine Learning*, 8748–8763. PMLR.

Sadeghian, A.; Armandpour, M.; Colas, A.; and Wang, D. Z. 2021. ChronoR: Rotation Based Temporal Knowledge Graph Embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7): 6471–6479.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*, 593–607. Cham: Springer International Publishing.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.; Shum, H.-Y.; and Guo, J. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR)*.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. GraphGPT: Graph Instruction Tuning for Large Language Models. In *Proceedings of the Forty-Seventh International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500. New York, NY, USA: Association for Computing Machinery.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Wang, D.; Zuo, Y.; Li, F.; and Wu, J. 2024a. LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings. In *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems*, 5950–5973. Curran Associates, Inc.

Wang, J.; Cui, Z.; Wang, B.; Pan, S.; Gao, J.; Yin, B.; and Gao, W. 2024b. IME: Integrating Multi-curvature Shared and Specific Embedding for Temporal Knowledge Graph Completion. In *Proceedings of the ACM Web Conference 2024*, 1954–1962. New York, NY, USA: Association for Computing Machinery.

Wang, J.; Sun, K.; LUO, L.; Wei, W.; Hu, Y.; Liew, A. W.-C.; Pan, S.; and Yin, B. 2024c. Large Language Models-guided Dynamic Adaptation for Temporal Knowledge Graph Reasoning. In *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems*, 8384–8410. Curran Associates, Inc.

Wen, Z.; and Fang, Y. 2023. Augmenting Low-Resource Text Classification with Graph-Grounded Pre-training and Prompting. In *Proceedings of the Forty-Sixth International ACM SIGIR Conference on Research and Development in Information Retrieval*, 506–516. New York, NY, USA: Association for Computing Machinery.

Xiong, S.; Yang, Y.; Fekri, F.; and Kerce, J. C. 2023. TILP: Differentiable Learning of Temporal Logical Rules on Knowledge Graphs. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical Evaluation of Rectified Activations in Convolutional Network. arXiv:1505.00853.

Xu, W.; Liu, B.; Peng, M.; Jia, X.; and Peng, M. 2023. Pre-trained Language Model with Prompts for Temporal Knowledge Graph Completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, 7790–7803. Toronto, Canada: Association for Computational Linguistics.

Yang, B.; Yih, S. W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*.

Zhang, M.; Sun, M.; Wang, P.; Fan, S.; Mo, Y.; Xu, X.; Liu, H.; Yang, C.; and Shi, C. 2024a. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM Web Conference 2024*, 1003–1014. New York, NY, USA: Association for Computing Machinery.

Zhang, Y.; Chen, Z.; Guo, L.; Xu, Y.; Zhang, W.; and Chen, H. 2024b. Making Large Language Models Perform Better in Knowledge Graph Completion. In *Proceedings of the Thirty-Second ACM International Conference on Multimedia*, 233–242. New York, NY, USA: Association for Computing Machinery.