Leveraging Implicit Contexts via LLM–Graph Fusion for Temporal Knowledge Graph Reasoning

Zeshu Tian[1], **Junjie Tao**[1], **Hongli Zhang**[1]
[1]Harbin Institute of Technology, Harbin, China
tianzeshu@hit.edu.cn, taojunjie@stu.hit.edu.cn, zhanghongli@hit.edu.cn

Abstract

Temporal knowledge graph (TKG) reasoning is critical for modeling and forecasting the evolution of real-world events. Existing TKG construction pipelines transform raw text into structured temporal quadruples as graph facts. However, in this process, they often fail to preserve reasoning-relevant contextual semantics from the original corpus that cannot be explicitly represented as graph facts, leaving such implicit contextual information unused by current temporal reasoning models. To address this limitation, we propose a Textual-Temporal Graph Fusion Network (TTGFN), a context-aware framework that leverages implicit contexts from text via LLM-based semantic encoding and fuses it with structure-constrained temporal graph representations for reasoning. To the best of our knowledge, this is the first work to systematically leverage LLMs to reuse previously overlooked implicit contextual information and incorporate it into temporal knowledge graph reasoning, substantially improving model performance. Extensive experiments conducted on three TKG reasoning benchmark datasets demonstrate that TTGFN outperforms the state-of-the-art approaches, with Hits@1 gains of 20.85% on ICEWS14 dataset, 31.14% on ICEWS05-15 dataset, and 24.22% on ICEWS18 dataset, respectively.

1 Introduction

Temporal Knowledge Graph (TKG) reasoning aims to model and predict how real-world events evolve over time, and it underpins many time-critical forecasting tasks such as stock market prediction [Zhu *et al.*, 2025], political stability assessment [Gwak *et al.*, 2024], and crisis response planning [Chen *et al.*, 2024], where future outcomes depend on the structured progression of past events. In practice, most TKGs are built from raw text through an information extraction pipeline that converts event mentions into structured temporal facts, typically represented as *(subject, relation, object, timestamp)* quadruples [Cai *et al.*, 2023a].

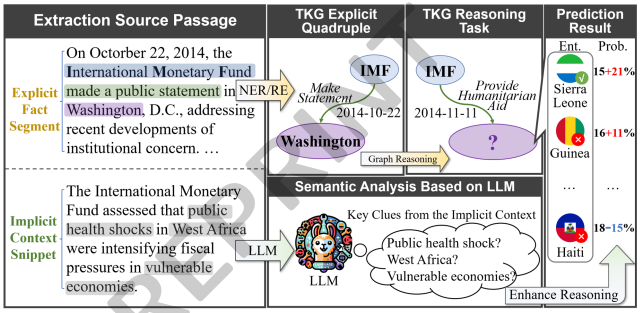


Figure 1: Motivation of our approach. Traditional TKG reasoning relies solely on explicit facts, leading to limited query representations, while leveraging implicit contexts enhances temporal prediction.

However, this pipeline inherently limits what evidence can be used for reasoning. It retains only text that can be explicitly extracted into a fixed quadruple form (e.g., via Named Entity Recognition or Relation Extraction). *We term such extractable textual evidence as explicit context.* Many informative cues in the source passage, such as explanations, thematic signals, and regional conditions, cannot be cast into a single quadruple. *We term such non-extractable yet informative evidence as implicit contexts.* Prior studies on automatic knowledge graph construction have observed that extraction-based pipelines often fail to preserve reasoning-relevant implicit contextual semantics from the original corpus [Zhong *et al.*, 2023], leaving them unused by structure-only TKG reasoning models and limiting predictive performance in complex scenarios. Fig. 1 illustrates how implicit cues can constrain subsequent temporal reasoning queries. Therefore, acquiring and utilizing implicit contexts for TKG reasoning remains an important problem.

To enable effective acquisition and utilization of implicit contexts in TKG reasoning, we need to address the following significant challenges:

CH1. How can implicit contexts be systematically acquired and effectively utilized to enhance query representations for temporal prediction? The first challenge concerns how to reliably identify, acquire and distill useful implicit signals from unstructured text for a given temporal query. Unlike explicit contexts that are directly recorded as

graph facts, implicit contexts are scattered, heterogeneous, and often noisy, with relevance that varies across time and queries. Effectively exploiting such information therefore requires temporally aligned context selection as well as compact and noise-robust representation learning, so that the distilled semantics can consistently strengthen temporal link prediction.

CH2. How can semantic representations from implicit contexts be fused with structure-constrained temporal representations? The second challenge concerns coherently fusing semantic representations from implicit contexts with graph-based temporal representations that are constrained by both graph structure and temporal dynamics. Temporal graph models produce structure-aware embeddings by explicitly encoding temporal and relational dependencies, whereas representations derived from implicit text are typically high-dimensional and lack explicit structural constraints [Cai *et al.*, 2023b]. Bridging this gap requires a fusion mechanism that respects structural and temporal constraints, ensuring that implicit semantic information complements graph-based reasoning and leads to stable temporal predictions.

Our Approach and Contributions. To address these challenges, we propose the **T**extual-**T**emporal **G**raph **F**usion **N**etwork (TTGFN), a framework that enhances TKG reasoning by acquiring implicit contexts from text and integrating its LLM-encoded representations with graph-based query representations. Rather than augmenting the graph with additional extracted facts or modifying its structure, TTGFN enhances the original prediction query through context-aware semantic fusion. The main contributions of our work are summarized as follows:

- To address **CH1**, we propose a context-aware inference module that acquires implicit contexts and encodes it into compact semantic priors using LLMs, thereby enhancing query representations for temporal link prediction.
- To address **CH2**, we propose a structure-constrained fusion mechanism that aligns LLM-derived semantic representations with temporal graph embeddings, enabling stable and synergistic enhanced reasoning.
- Extensive experiments on three TKG reasoning benchmarks demonstrate that TTGFN achieves state-of-the-art performance. We also release three implicit-context-augmented benchmark datasets[1] to facilitate future research in the community.

2 Related Work

2.1 Temporal Knowledge Graph Reasoning

Recent advances in TKG reasoning have been largely driven by neural temporal modeling over observed quadruples. Representative methods capture historical dependencies through autoregressive or recurrent dynamics (e.g., RE-NET [Jin *et al.*, 2019]), incorporate relation-aware temporal message passing and updates (e.g., REGCN [Li *et al.*, 2021a]), and further enhance temporal modeling by combining local and global temporal patterns (e.g., TiRGN [Li *et al.*, 2022]). More recent work explores richer temporal signals, such as trend-aware representations for evolving event dynamics (Trend [Huang *et al.*, 2024]), or formulates temporal reasoning as long-horizon decision making to improve dependency modeling (LogiRL [Chen *et al.*, 2025a]). Despite these advances, most existing methods still operate primarily on structured quadruples and do not systematically exploit the rich textual context available during graph construction, limiting their ability to leverage implicit background knowledge for temporal prediction.

2.2 Integration of LLMs with TKGs

Recent studies have explored integrating LLMs with graph-based models for prediction and reasoning [Li *et al.*, 2023b; Jin *et al.*, 2024]. In TKG settings, most approaches use LLMs as auxiliary components, such as generating temporal rules, modeling logical constraints, or reformulating forecasting as language-based inference [Wang *et al.*, 2024a; Bai *et al.*, 2025a; Pan *et al.*, 2025]. However, they often overlook leveraging implicit textual context that is discarded during graph construction to systematically strengthen the prediction query. In contrast, our work reuses such implicit contexts and aligns their semantic representations with structure-constrained temporal representations for more expressive TKG reasoning.

3 Problem Definition

A TKG is denoted as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{Q}\}$, where $\mathcal{E}$ is the entity set, $\mathcal{R}$ is the relation set, $\mathcal{T}$ is the timestamp set, and $\mathcal{Q}$ is the set of temporal quadruples. Each quadruple is represented as $(e_s, r, e_o, t) \in \mathcal{Q}$, where $e_s, e_o \in \mathcal{E}$ denote the head and tail entities, $r \in \mathcal{R}$ denotes the relation, and $t \in \mathcal{T}$ denotes the timestamp. Given a query (e_s, r, t), the task of TKG reasoning is formulated as predicting the tail entity e_o.

4 Methodology

Fig. 2 illustrates our framework for temporal link prediction as a unified three-stage pipeline. Stage I retrieves implicit context snippets for the query entity using reproducible strategies. Stage II encodes the retrieved context and query information into dense, context-aware representations via a LLM. Stage III integrates these representations with graph-guided reasoning to predict target entities under temporal and structural constraints.

4.1 Stage I: Implicit Context Acquisition

In this work, we define implicit contexts as auxiliary semantic information from unstructured or weakly structured textual sources related to an entity that is not explicitly captured by knowledge graph quadruples, yet remains informative for temporal reasoning. This stage aims to recover such contextual snippets in a concise and reproducible manner.

Implicit Context Repository. We construct an offline *Implicit Context Repository* (ICR) that stores context snippets

[1]Our code and three implicit-context-augmented datasets are publicly available at https://github.com/tianzeshu/TTGFN.

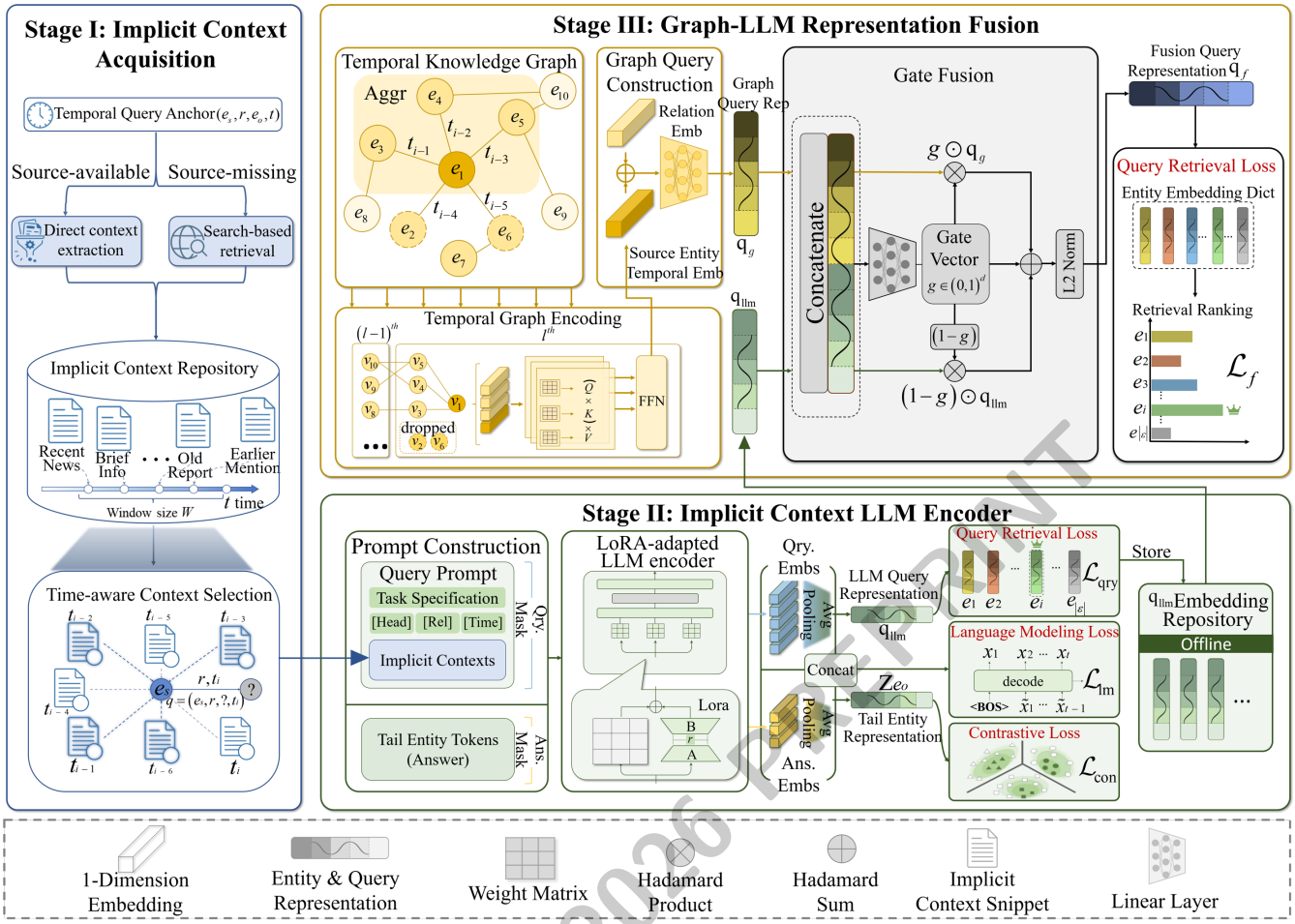


Figure 2: The overall framework of TTGFN.

associated with each query anchor (e_s, t) and temporally preceding t (e.g., within $[t-w, t)$, where w denotes a temporal window size), obtained either from available unstructured texts or via entity-anchored retrieval from external corpora.

Each snippet is represented as a pair (t'_j, x_j), where x_j is a natural language snippet related to e_s and $t'_j \in \mathcal{T}$ denotes its timestamp; such snippets do not necessarily mention the target entity or queried relation and may contain noise. For each anchor (e_s, t), snippets are as follows:

$$\mathcal{M}[e_s, t] = \{(t'_j, x_j)\}_{j=1}^{N_{e_s,t}}, \tag{1}$$

where $N_{e_s,t}$ denotes the number of snippets. Each snippet is indexed by (e_s, t'_j), enabling entity-conditioned and time-aware organization, while the coexistence of informative and irrelevant snippets is handled in Stage II via a two-step training strategy (Section 4.2).

Time-aware Context Selection. At training or inference time, given a temporal query (e_s, r, t), we retrieve its associated snippet set $\mathcal{M}[e_s, t]$ from the ICR and select only snippets whose timestamps precede the query time:

$$\mathcal{C}_{e_s,t} = \{(t'_j, x_j) \in \mathcal{M}[e_s, t] \mid t'_j < t\}. \tag{2}$$

To emphasize temporal relevance, we retain the K snippets closest to t:

$$\mathcal{C}^{(K)}_{e_s,t} = \textsc{TopK}(\mathcal{C}_{e_s,t}). \tag{3}$$

This design prevents temporal leakage and preserves the most relevant implicit contexts for subsequent semantic encoding, where $\textsc{TopK}(\cdot)$ ranks snippets by their timestamps.

4.2 Stage II: Implicit Context LLM Encoder

Stage II encodes implicit contexts associated with each temporal query into compact, reusable semantic representations using a large language model, providing context-aware information for downstream graph-based temporal reasoning.

Prompt Construction

For each temporal query (e_s, r, t), we construct a task-oriented prompt by instantiating a fixed template with the query tuple and the retrieved implicit contexts:

$$P^{\text{ictx}}_{e_s,r,t} = \text{Prompt}(e_s, r, t; \mathcal{C}^{(K)}_{e_s,t}), \tag{4}$$

where superscripts ictx indicate whether implicit context (ictx) snippets are included in the prompt.

During training, we follow a teacher-forced formulation and append the ground-truth tail entity text to the query

prompt, yielding an answer-augmented input $P^{\text{ictx}}_{e_s,r,t,e_o}$, while inference and downstream fusion use $P^{\text{ictx}}_{e_s,r,t}$ alone.

Context-Enhanced Representation Generation

Given the input prompt $\hat{P}$ (i.e., P_{e_s,r,t,e_o} for training and $P_{e_s,r,t}$ for inference), the LoRA-adapted encoder f_θ produces token-level representations:

$$\mathbf{H} = f_\theta(\hat{P}) \in \mathbb{R}^{L\times d}. \tag{5}$$

Let $\mathcal{I}_{\text{qry}}, \mathcal{I}_{\text{ans}} \subseteq \{1,\dots,L\}$ denote the token index sets of the query and answer segments, respectively. Segment representations are obtained via masked mean pooling:

$$\mathbf{q}_{\text{llm}} = \frac{1}{|\mathcal{I}_{\text{qry}}|}\sum_{\ell\in\mathcal{I}_{\text{qry}}}\mathbf{H}_\ell, \qquad \mathbf{z}_{e_o} = \frac{1}{|\mathcal{I}_{\text{ans}}|}\sum_{\ell\in\mathcal{I}_{\text{ans}}}\mathbf{H}_\ell. \tag{6}$$

Importantly, $\mathbf{q}_{\text{llm}}$ is computed exclusively from the query segment to prevent label leakage and ensure train–test consistency.

Multi-Objective Optimization

We optimize the prompt encoder using a multi-objective formulation to handle noisy and weakly aligned implicit contexts.

Query Retrieval Loss. Given the context-aware query representation $\mathbf{q}_{\text{llm}}$ and a fixed entity embedding dictionary $\mathbf{E}$ that stores embeddings for all candidate entities $e \in \mathcal{E}$, pre-encoded by the same base language model used for prompt encoding:

$$\mathbf{E} = [\mathbf{e}_{e_1},\dots,\mathbf{e}_{e_{|\mathcal{E}|}}]^\top, \tag{7}$$

we compute retrieval logits $s(e) = \mathbf{q}_{\text{llm}}^\top \mathbf{e}_e/\tau_q$ for $e \in \mathcal{E}$, where $\tau_q > 0$ is a temperature parameter, and define the retrieval loss as follows:

$$\mathcal{L}_{\text{qry}} = -\log \frac{\exp\big(\mathbf{q}_{\text{llm}}^\top \mathbf{e}_{e_o}/\tau_q\big)}{\sum_{e\in\mathcal{E}}\exp\big(\mathbf{q}_{\text{llm}}^\top \mathbf{e}_e/\tau_q\big)}. \tag{8}$$

Language Modeling Loss. Under the teacher-forced formulation adopted during training, let $P_{e_s,r,t,e_o} = (x_1,\dots,x_L)$ denote the answer-augmented prompt, where x_ℓ is the ℓ-th input token. We apply a language modeling loss only on tokens corresponding to the appended ground-truth tail entity, indexed by $\mathcal{I}_{\text{ans}}$ to regularize hidden representations under retrieval supervision as follows:

$$\mathcal{L}_{\text{lm}} = -\sum_{\ell\in\mathcal{I}_{\text{ans}}}\log p_\theta(x_\ell \mid x_{<\ell}, P_{e_s,r,t,e_o}), \tag{9}$$

where $p_\theta(\cdot)$ denotes the conditional token probability induced by the language model with parameters θ.

Contrastive Loss. To encourage the query representation to selectively attend to context cues that are discriminative for identifying the correct tail entity, we introduce a contrastive objective between the query representation $\mathbf{q}_{\text{llm}}$ and the tail entity representation $\mathbf{z}_{e_o}$ extracted from the same forward computation. For a mini-batch of size B, the contrastive loss is formulated as:

$$\mathcal{L}_{\text{con}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(a_{ii})}{\sum_{j=1}^{B}\exp(a_{ij})},\ a_{ij} = \frac{\big(\mathbf{q}^i_{\text{llm}}\big)^\top \mathbf{z}^j_{e_o}}{\tau_c}, \tag{10}$$

Algorithm 1: Stage II Training Procedure

Input: Temporal training queries $\mathcal{Q}$, implicit contexts index $\mathcal{M}$
Output: Trained encoder f_θ

Step 1: Train with context-free prompts;
foreach $(e_s, r, t, e_o) \in \mathcal{Q}$ **do**
  Construct the answer-augmented prompt P_{e_s,r,t,e_o};
  Compute $\mathbf{H} = f_\theta(P_{e_s,r,t,e_o})$ (Eq. (5));
  Obtain $\mathbf{q}_{\text{llm}}$ and $\mathbf{z}_{e_o}$ by masked pooling (Eq. (6));
  Compute $\mathcal{L}_{\text{qry}}, \mathcal{L}_{\text{lm}}, \mathcal{L}_{\text{con}}$, and optimize $\mathcal{L}_{\text{llm}}$;
Step 2: Fine-tune with implicit contexts;
foreach $(e_s, r, t, e_o) \in \mathcal{Q}$ **do**
  Retrieve $\mathcal{C}^{(K)}_{e_s,t}$ from $\mathcal{M}$ and construct the answer-augmented ictx-aware prompt $P^{\text{ictx}}_{e_s,r,t,e_o}$;
  Compute $\mathbf{H} = f_\theta(P^{\text{ictx}}_{e_s,r,t,e_o})$ (Eq. (5));
  Obtain $\mathbf{q}_{\text{llm}}$ and $\mathbf{z}_{e_o}$ by masked pooling (Eq. (6));
  Compute $\mathcal{L}_{\text{qry}}, \mathcal{L}_{\text{lm}}, \mathcal{L}_{\text{con}}$, and optimize $\mathcal{L}_{\text{llm}}$;
return f_θ;

where a_{ij} denotes the batch-wise similarity between the i-th LLM-based query representation $\mathbf{q}^i_{\text{llm}}$ and the j-th tail entity representation $\mathbf{z}^j_{e_o}$, and $\tau_c > 0$ is a temperature parameter.

Overall Objective. The overall training objective of Stage II is defined as follows:

$$\mathcal{L}_{\text{llm}} = \mathcal{L}_{\text{qry}} + \lambda_{\text{lm}}\mathcal{L}_{\text{lm}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}, \tag{11}$$

where λ_{lm} and λ_{con} control the contributions of semantic regularization and context disambiguation, respectively.

Two-Step Training Strategy

Implicit context snippets are weakly aligned and may contain noise, and directly incorporating them from the beginning of training can cause the model to overfit spurious correlations. To mitigate this issue, Stage II adopts a two-step training strategy based on whether implicit contexts are included. The overall training and inference procedure of Stage II is summarized in Algorithm 1.

Specifically, we first train the model with context-free prompts that exclude implicit contexts, allowing the model to focus on the core temporal prediction task without interference from noisy contextual signals. We then fine-tune the model using context-aware prompts $P^{\text{ictx}}_{e_s,r,t}$ with $\mathcal{C}^{(K)}_{e_s,t}$ to selectively incorporate implicit contextual semantics.

4.3 Stage III: Graph-LLM Representation Fusion

We treat the temporal graph encoder as a structure- and time-constrained semantic module that complements the language-based query representation with inductive bias. Entities and relations are embedded in the same semantic space as the entity dictionary $\mathbf{E}$, where both entities and relations are initialized by projecting their LLM-induced embeddings, yielding the initial entity embedding $\mathbf{v}^{(0)}_e$ and the relation embedding $\mathbf{v}_r$, respectively.

Temporal Graph Encoding. Given a temporal query (e_s, r, t), the temporal neighborhood $\mathcal{N}(e_s, t)$ consists of all historical interactions involving e_s that occurred strictly before t, each represented as $(e_u, r_{u\to s}, t_{u\to s})$, where $e_u \in \mathcal{E}$ is a neighboring entity, $r_{u\to s} \in \mathcal{R}$ denotes the relation type, and $t_{u\to s} \in \mathcal{T}$ is the interaction timestamp. To capture temporal dynamics, the time gap between each interaction and t is encoded using a continuous time encoding function $\boldsymbol{\tau}(\cdot)$. Following TGAT [Xu *et al.*, 2020], relation-aware and time-conditioned messages are aggregated from $\mathcal{N}(e_s, t)$, with each message defined as:

$$\mathbf{m}_{u\to s,t} = \mathbf{W}_{\text{msg}}\big[\mathbf{v}^{(\ell-1)}_{e_u,t_{u\to s}} \,;\, \mathbf{v}_{r_{u\to s}} \,;\, \boldsymbol{\tau}(t - t_{u\to s})\big], \quad (12)$$

where $\mathbf{W}_{\text{msg}}$ is a learnable weight matrix.

After L_g aggregation layers, the head entity temporal embedding $\mathbf{v}^{(L_g)}_{e_s,t}$ is combined with the query relation embedding $\mathbf{v}_r$ to form the graph-side semantic query, defined as:

$$\mathbf{q}_g = \phi\big([\mathbf{v}^{(L_g)}_{e_s,t}; \mathbf{v}_r]\big), \quad (13)$$

where $\phi(\cdot)$ denotes a linear projection followed by normalization; $\mathbf{q}_g$ is subsequently used for gated fusion with the language-based query.

Gated Fusion of Language and Graph Queries. The language-based query $\mathbf{q}_{\text{llm}}$ and the graph-based query $\mathbf{q}_g$ are implicitly aligned through shared supervision over the entity embedding dictionary $\mathbf{E}$ and consistent semantic initialization, yielding representations in a common semantic space. They are combined via a gated fusion mechanism, where the gate vector $\mathbf{g}$ controls the dimension-wise contribution of each query. The fused query representation is obtained as:

$$\mathbf{g} = \sigma\big([\mathbf{q}_{\text{llm}}; \mathbf{q}_g]\big), \quad (14)$$

$$\mathbf{q}_f = \text{Norm}\Big(\mathbf{g} \odot \mathbf{q}_{\text{llm}} + (1 - \mathbf{g}) \odot \mathbf{q}_g\Big), \quad (15)$$

where $\text{Norm}(\cdot)$ denotes ℓ_2 normalization, and $\odot$ denotes element-wise multiplication.

Predictions are made by ranking candidate entities according to $\mathbf{q}_f^\top \mathbf{e}_e$, and the model is trained using a softmax-based query loss:

$$\mathcal{L}_f = -\log \frac{\exp\Big(\mathbf{q}_f^\top \mathbf{e}_{e_o}/\tau\Big)}{\sum_{e\in\mathcal{E}} \exp\Big(\mathbf{q}_f^\top \mathbf{e}_e/\tau\Big)}. \quad (16)$$

5 Experiments

We conduct extensive experiments to answer the following research questions (RQs):

- **RQ1**: How does TTGFN perform when compared with state-of-the-art methods on TKG reasoning tasks?
- **RQ2**: How do individual components of TTGFN contribute to overall performance and robustness against hallucinated contextual signals?
- **RQ3**: How does the quality of implicit contexts, in terms of relevance and temporal proximity, affect TTGFN and its sensitivity to hallucinated contexts?

Statistic	ICEWS14	ICEWS05–15	ICEWS18
#Entities	7,128	10,488	23,033
#Relations	230	251	256
#Timestamps	365	4,017	365
#Samples	90,706	461,200	465,538
#Granularity	24 hours	24 hours	1 hour

Table 1: Statistics of the ICEWS datasets used in our experiments.

- **RQ4**: How robust is TTGFN under long-tail relations and long-range temporal reasoning?
- **RQ5**: How sensitive is TTGFN to loss-weighting parameters balancing semantic modeling and structure-constrained alignment?

5.1 Experimental Settings

Datasets

We conduct experiments on three widely used ICEWS TKG benchmarks[2], which record fine-grained socio-political events as temporal quadruples. Specifically, we use ICEWS14, ICEWS05–15, and ICEWS18, following standard 80%/10%/10% train/validation/test splits. All models are evaluated on the link prediction task, with dataset statistics summarized in Table 1. Additional dataset statistics, including temporal coverage, graph scale and three implicit-context-augmented datasets constructed in this work, are provided in Appendix Section 3.

Evaluation Metrics

We evaluate performance using standard ranking-based metrics, including Mean Reciprocal Rank (MRR) and Hits@k for $k \in \{1, 3\}$. We focus on temporal link prediction, where the model predicts the tail entity o given $(s, r, ?, t)$. Following common practice [Cai *et al.*, 2023b], we adopt the time-aware filtered evaluation protocol provided by xERTE [Han *et al.*, 2021] to ensure fair and consistent evaluation.

Parameter Settings

Experiments are conducted on a single NVIDIA RTX 5090 GPU, while the codebase also supports multi-GPU training via Distributed Data Parallel (DDP). Unless otherwise stated, all runs are executed with fixed random seeds to ensure reproducibility.

We train TTGFN with AdamW (initial learning rate 1×10^{-4}) for up to 50 epochs and apply early stopping (patience $= 3$) based on time-aware filtered validation MRR. To avoid frequent checkpoint updates caused by marginal fluctuations, we discretize the monitored MRR by flooring to 10^{-3} precision. We optionally use gradient accumulation and a cosine annealing learning-rate schedule for stable optimization. This discretization stabilizes early stopping without affecting final rankings. All results are averaged over five independent runs to ensure statistical robustness. Detailed parameter settings are provided in Appendix Section 1.

[2]https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075

Method	ICEWS14			ICEWS05-15			ICEWS18		
	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
Static Graph-based Methods									
Conv-TransE [Shang *et al.*, 2019]	31.50	22.46	34.98	30.28	20.79	33.80	23.22	14.26	26.13
ConvE [Liu *et al.*, 2022]	30.30	21.30	34.42	31.40	21.56	35.70	22.81	13.63	25.83
Temporal Graph-based Methods									
RGCRN [Seo *et al.*, 2018]	33.31	24.08	36.55	33.00	24.00	36.50	23.46	14.24	26.62
ReNet [Jin *et al.*, 2020]	35.77	25.99	40.41	36.86	26.24	41.85	26.17	16.43	29.89
RE-GCN [Li *et al.*, 2021b]	41.50	30.86	46.60	46.41	35.17	52.76	30.55	20.00	34.73
CYGNET [Zhu *et al.*, 2021]	35.09	25.57	39.96	36.20	26.15	41.37	25.09	15.96	28.75
TiRGN [Li *et al.*, 2022]	43.88	33.12	49.48	48.72	37.17	55.48	32.06	21.08	36.75
STSP [Li *et al.*, 2023a]	41.38	30.62	46.96	35.87	25.43	37.50	30.93	20.30	35.20
CENet [Xu *et al.*, 2023b]	38.19	28.46	42.81	41.04	30.47	46.51	28.03	18.09	32.09
Trend [Huang *et al.*, 2024]	46.33	36.01	51.76	44.21	32.78	49.86	34.67	24.02	39.30
LogiRL [Chen *et al.*, 2025a]	44.71	35.72	51.03	51.04	40.71	57.55	34.94	24.76	39.57
LLM-based Methods									
PPT [Xu *et al.*, 2023a]	38.42	28.94	42.50	38.85	28.57	43.35	26.63	16.94	30.64
GenTKG [Liao *et al.*, 2024]	-	36.85	47.95	-	-	-	-	24.25	37.25
G2S [Bai *et al.*, 2025b]	-	38.33	54.12	-	-	-	-	23.04	35.32
AnRe [Tang *et al.*, 2025]	47.40	36.90	51.10	50.90	39.10	58.00	35.50	26.00	39.20
Graph-LLM Hybrid Methods									
TiRGN + CoH [Xia *et al.*, 2024]	43.94	33.07	49.64	49.71	38.01	56.40	32.98	21.83	37.79
LLM-DA [Wang *et al.*, 2024b]	47.10	36.90	<u>52.60</u>	52.10	41.60	58.60	-	-	-
DSEP [Deng *et al.*, 2025]	44.87	34.35	50.29	50.68	39.78	56.85	33.81	23.24	38.22
CognTKE [Chen *et al.*, 2025b]	46.06	36.49	51.11	<u>53.13</u>	<u>42.62</u>	<u>59.42</u>	35.24	25.21	<u>39.93</u>
APE-LLaMA-7B [Yang and Zhu, 2025]	<u>48.32</u>	<u>37.74</u>	-	52.86	41.94	-	<u>36.52</u>	<u>26.26</u>	-
Our Method									
TTGFN	**52.58**	**45.61**	**55.94**	**61.28**	**55.89**	**64.00**	**38.58**	**32.62**	**40.90**
Δ Improvement	**+8.82%**	**+20.85%**	**+6.35%**	**+15.34%**	**+31.14%**	**+7.71%**	**+5.64%**	**+24.22%**	**+2.37%**

Table 2: Link prediction results (%). Bold denotes the best overall performance, while underlined values indicate the best results among baseline methods. Δ Improvement reports the relative gain of TTGFN over the best-performing baseline for each metric.

Baselines for Comparison

To comprehensively evaluate the effectiveness of TTGFN, we compare it with representative baseline methods summarized in Table 2. All baselines are evaluated using publicly available implementations or by following the experimental settings reported in their original papers. Detailed descriptions of the baseline methods are provided in Appendix Section 2.

5.2 Result 1: Comparison with Baselines

To answer **RQ1**, we evaluate TTGFN on temporal link prediction over ICEWS14, ICEWS05–15, and ICEWS18, with results reported in Table 2. Overall, TTGFN consistently outperforms all baseline methods across all datasets and all Hits@k metrics, demonstrating robust and comprehensive improvements in ranking performance. In particular, TTGFN achieves gains of 8.82% in MRR and 20.85% in Hits@1 on ICEWS14, and achieves relative gains of up to 15.34% in MRR and 31.14% in Hits@1 on ICEWS05–15, as well as 5.64% in MRR and 24.22% in Hits@1 on ICEWS18, indicating more accurate and reliable tail entity prediction.

These consistent gains suggest that reusing implicit contextual information primarily enhances the discriminative power of query representations, rather than merely reshaping the ranking distribution in isolated cases. By integrating semantic cues recovered by LLMs with structure-constrained temporal representations, TTGFN produces more informative query embeddings, leading to improved top-ranked prediction accuracy across diverse TKGs. We also provide a detailed analysis of training and inference efficiency in Appendix Section 6.

Variant	ICEWS14		ICEWS05-15		ICEWS18	
	MRR	H@1	MRR	H@1	MRR	H@1
TTGFN	**52.58**	**45.61**	**61.28**	**55.89**	**38.58**	**32.62**
LLM-only	49.13	42.54	56.44	50.53	34.91	28.65
Graph-only	47.07	40.69	58.45	53.05	33.14	27.21
w/o ICR	48.02	41.58	59.36	53.92	34.02	27.88
AvgFusion	50.11	43.27	60.02	54.41	35.96	30.04
ScalarGate	51.34	44.39	60.71	55.02	37.12	31.06

Table 3: Link prediction results of TTGFN variants.

5.3 Result 2: Ablation Study

To answer **RQ2**, we conduct an ablation study under identical hyper-parameter settings, with results summarized in Table 3. This experiment evaluates the contribution of four key designs in TTGFN: language-based semantic reasoning, temporal graph reasoning, implicit context retrieval, and gated representation fusion. We compare the full TTGFN with five ablated variants. Specifically, *LLM-only* removes the graph-

based reasoning path and relies on language-derived representations, while *Graph-only* disables the language branch and performs reasoning only on temporal graph structure. The *w/o ICR* variant removes implicit context retrieval and forces the model to operate without external textual context. To examine the fusion design, *AvgFusion* replaces the gated fusion module with uniform averaging of the two query representations, whereas *ScalarGate* adopts a simplified fusion mechanism using a learnable scalar gate.

The results show that removing any core component consistently degrades performance across the three datasets. The declines of *LLM-only* and *Graph-only* indicate that language-derived semantics and temporal graph structure provide complementary evidence for temporal link prediction. The degradation of *w/o ICR* further confirms that implicit context retrieval contributes useful external semantic information beyond explicit temporal quadruples. Moreover, although *AvgFusion* and *ScalarGate* still combine both representation sources, they remain inferior to the full TTGFN, suggesting that simple averaging or scalar-level weighting cannot fully capture fine-grained interactions between graph-side and language-side signals.

These ablation results demonstrate that TTGFN improves temporal reasoning through coordinated semantic enhancement and structure-constrained fusion, rather than by simply introducing LLM-encoded features. They also address the hallucination concern associated with LLM-based semantics, since TTGFN consistently outperforms both the *LLM-only* and naive fusion variants. This suggests that semantic cues are injected in a controlled manner, while the gated fusion mechanism helps suppress spurious or hallucinated signals through temporal graph constraints.

5.4 Result 3: Analysis of Implicit Context Quality

To answer **RQ3**, we analyze the impact of implicit context quality on temporal link prediction using a binned evaluation (Fig. 3). Implicit context snippets are grouped by their average time gap Δt and LLM-estimated relevance score ρ, aggregated over the test queries that select each context. This setting allows us to examine how temporal proximity and semantic relevance jointly affect the usefulness of retrieved implicit contexts. Performance gains are measured as ΔMRR and ΔHits@1 relative to the no-context setting.

Across all datasets, implicit contexts consistently improve performance, with larger gains for contexts that are temporally closer and more semantically relevant. This trend demonstrates that TTGFN can effectively exploit high-quality implicit contexts to enhance temporal query representations. Meanwhile, the performance gains become smaller when contexts are less relevant or temporally distant, indicating that the model remains robust under limited relevance or temporal proximity rather than being overly sensitive to noisy contextual information.

Regarding hallucination, the binned analysis indicates that improvements concentrate in temporally proximate and high-relevance contexts, while low-relevance contexts yield negligible impact. This selectivity suggests that TTGFN does not blindly amplify contextual semantics and is relatively robust to unreliable or hallucinated signals. Overall, the analysis

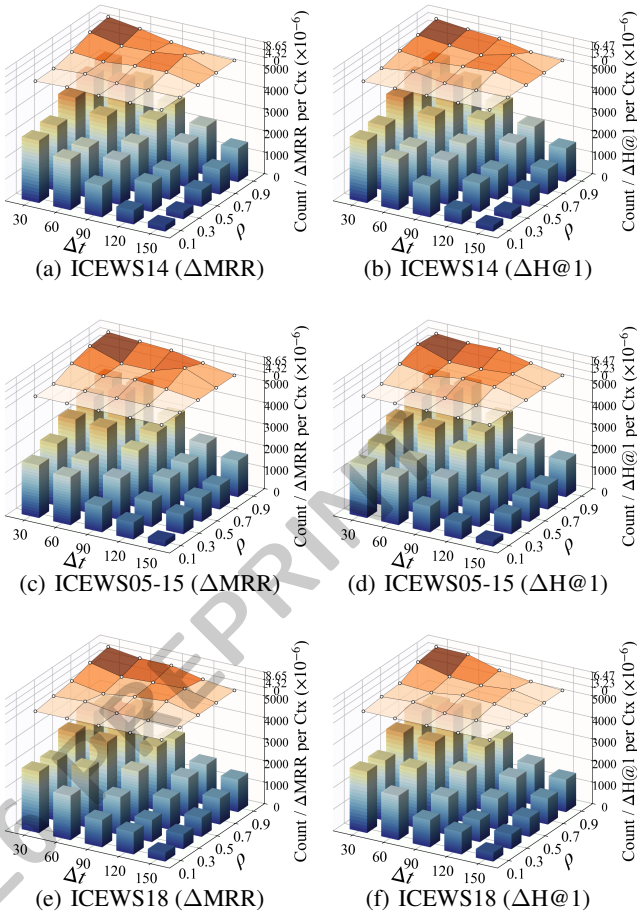


Figure 3: Binned analysis of implicit context effectiveness on temporal link prediction. Bins are defined by the time gap Δt and LLM-estimated relevance ρ. 3D columns denote instance counts, and surfaces show performance gains (ΔMRR on the left and ΔHits@1 on the right) on ICEWS14, ICEWS05–15, and ICEWS18.

supports that implicit contexts are beneficial when they provide relevant temporal evidence, while low-quality contexts have limited influence on prediction. Detailed analyses are provided in Appendix Section 5.

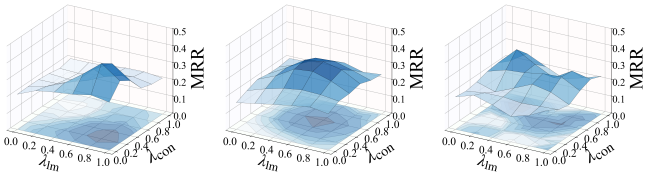
5.5 Result 4: Long-Tail and Long-Range Reasoning Analysis

To answer **RQ4**, we analyze the robustness of TTGFN under long-tail relations and long-range temporal dependencies, with results reported in Table 4. Test queries are analyzed along two dimensions: relation frequency and temporal distance to the most recent historical observation. For relation frequency, relation types are ranked by occurrence frequency and split into three groups with similar numbers of relation types. For temporal distance, queries are grouped into temporal-gap tertiles based on the empirical distribution, with no-history queries considered separately.

Across both analyses, TTGFN maintains relatively stable MRR as relation frequency decreases or temporal gaps increase. In the relation-frequency analysis, the model still

Tertile		ICEWS14		ICEWS05-15		ICEWS18	
		#Q	MRR	#Q	MRR	#Q	MRR
Freq.	Low	20	51.70	92	53.17	89	36.24
	Mid	232	52.45	967	57.93	1123	37.46
	High	8711	57.54	45033	61.08	48333	38.61
Δt	Short	2413	54.12	13914	62.34	17779	40.06
	Mid	2039	51.27	13278	60.58	11372	38.06
	Long	2178	48.18	13456	55.82	13898	37.36
	NH	2333	45.83	5444	53.18	6496	37.07

Table 4: Robustness analysis under relation frequency and temporal distance (RQ4). Relations are grouped into frequency tertiles (low, mid, and high), and queries are grouped by temporal distance into gap tertiles (short, mid, and long), with NH denoting no-history queries. We report the number of queries (#Q) and MRR (%).



(a) ICEWS14 (b) ICEWS05-15 (c) ICEWS18

Figure 4: Loss-weight sensitivity analysis of TTGFN (RQ5). MRR surfaces under different loss-weight settings on three datasets.

achieves reasonable performance on low-frequency relations, indicating its ability to handle long-tail reasoning cases despite limited observations. In the temporal-distance analysis, TTGFN shows moderate degradation from short-gap to long-gap settings, and it remains effective in no-history scenarios where recent structural evidence is unavailable.

These results demonstrate that TTGFN is robust under sparse structural supervision. Its performance in long-tail, long-range, and no-history settings suggests that the model can still make reliable predictions when explicit graph evidence becomes limited. This observation is consistent with the role of implicit contextual semantics in complementing sparse structural information, while the temporal graph component provides necessary structural constraints for prediction.

5.6 Result 5: Loss Weight Sensitivity Study

To address **RQ5**, we investigate the sensitivity of TTGFN to the loss-weight parameters λ_{lm} and λ_{con} in the overall Stage II training objective (Eq. (11)). These two parameters control the contributions of the language modeling loss and the contrastive loss, respectively. Fig. 4 shows the MRR performance surfaces on three datasets under different weight combinations.

Across all datasets, TTGFN exhibits a smooth performance landscape with a broad high-performance region, indicating low sensitivity to precise weight tuning. The model maintains strong performance under a range of moderate weight combinations, suggesting that the language modeling loss and contrastive loss can jointly improve the quality of LLM-based query representations. In contrast, extreme imbalance between λ_{lm} and λ_{con} causes moderate degradation, showing that overemphasizing either objective may weaken the overall representation quality.

These results demonstrate that TTGFN is robust to loss-weight selection and does not require precise dataset-specific parameter tuning. Balanced configurations consistently yield strong performance, supporting the joint use of language modeling supervision and contrastive learning in Stage II training. Accordingly, we adopt a balanced default setting $\lambda_{\text{lm}} = \lambda_{\text{con}}$ for all experiments, and fix both to 0.5 in practice unless otherwise specified.

6 Conclusion

In this paper, we propose TTGFN, which integrates LLMs with temporal graph networks for temporal knowledge graph reasoning. By aligning implicit contexts with structure-constrained temporal semantics, TTGFN enhances query representations without altering the graph structure. Experiments on ICEWS benchmarks show consistent gains, especially in long-tail and long-range settings.

References

[Bai *et al.*, 2025a] Long Bai, Zixuan Li, Xiaolong Jin, Jiafeng Guo, Xueqi Cheng, and Tat-Seng Chua. G2S: A general-to-specific learning framework for temporal knowledge graph forecasting with large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20927–20938, Vienna, Austria, July 2025. Association for Computational Linguistics.

[Bai *et al.*, 2025b] Long Bai, Zixuan Li, Xiaolong Jin, Jiafeng Guo, Xueqi Cheng, and Tat-Seng Chua. G2S: A general-to-specific learning framework for temporal knowledge graph forecasting with large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20927–20938, Vienna, Austria, July 2025. Association for Computational Linguistics.

[Cai *et al.*, 2023a] Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. Temporal knowledge graph completion: A survey. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6545–6553. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Survey Track.

[Cai *et al.*, 2023b] Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. Temporal knowledge graph completion: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6545–6553, 2023.

[Chen *et al.*, 2024] Minze Chen, Zhenxiang Tao, Weitong Tang, Tingxin Qin, Rui Yang, and Chunli Zhu. Enhancing emergency decision-making with knowledge graphs and

large language models. *International Journal of Disaster Risk Reduction*, 113:104804, 2024.

[Chen *et al.*, 2025a] Tingxuan Chen, Liu Yang, Zidong Wang, and Jun Long. A rule- and query-guided reinforcement learning for extrapolation reasoning in temporal knowledge graphs. *Neural Networks*, 185:107186, 2025.

[Chen *et al.*, 2025b] Wei Chen, Yuting Wu, Shuhan Wu, Zhiyu Zhang, Mengqi Liao, Youfang Lin, and Huaiyu Wan. Cogntke: a cognitive temporal knowledge extrapolation framework. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025.

[Deng *et al.*, 2025] Yunteng Deng, Jia Song, Zhongliang Yang, Yilin Long, Li Zeng, and Linna Zhou. Diachronic semantic encoding based on pre-trained language model for temporal knowledge graph reasoning. *Knowledge-Based Systems*, 318:113479, 2025.

[Gwak *et al.*, 2024] Daehoon Gwak, Junwoo Park, Minho Park, ChaeHun Park, Hyunchan Lee, Edward Choi, and Jaegul Choo. Forecasting future international events: A reliable dataset for text-based event modeling. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9000–9023, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[Han *et al.*, 2021] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*, 2021.

[Huang *et al.*, 2024] Shuxian Huang, Ye Wang, Kai Chen, and Yan Jia. Temporal relational context learning for extrapolation reasoning on temporal knowledge graphs. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6430–6434, 2024.

[Jin *et al.*, 2019] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*, 2019.

[Jin *et al.*, 2020] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, Online, November 2020. Association for Computational Linguistics.

[Jin *et al.*, 2024] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Li *et al.*, 2021a] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 408–417, 2021.

[Li *et al.*, 2021b] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 408–417, New York, NY, USA, 2021. Association for Computing Machinery.

[Li *et al.*, 2022] Yujia Li, Shiliang Sun, and Jing Zhao. Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *IJCAI*, pages 2152–2158, 2022.

[Li *et al.*, 2023a] Hongyang Li, Cui Zhu, Wenjun Zhu, Yunfeng Zhou, and Xinyu Zhang. Short-term sequential pattern for temporal knowledge graph reasoning. In *CAIBDA*, pages 728–739, 2023.

[Li *et al.*, 2023b] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.

[Liao *et al.*, 2024] Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. GenTKG: Generative forecasting on temporal knowledge graph with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4303–4317, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[Liu *et al.*, 2022] Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. Temporal knowledge graph reasoning via time-distributed representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 279–288, 2022.

[Pan *et al.*, 2025] Qihong Pan, Limin Yao, Guojiang Shen, Xiao Han, Yichuan Chen, and Xiangjie Kong. Leveraging temporal validity of rules via llms for enhanced temporal knowledge graph reasoning. *Knowledge-Based Systems*, 327:114094, 2025.

[Seo *et al.*, 2018] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa, editors, *Neural Information Processing*, pages 362–373, Cham, 2018. Springer International Publishing.

[Shang *et al.*, 2019] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3060–3067, Jul. 2019.

[Tang *et al.*, 2025] Guo Tang, Zheng Chu, Wenxiang Zheng, Junjia Xiang, Yizhuo Li, Weihao Zhang, Ming Liu, and Bing Qin. AnRe: Analogical replay for temporal knowledge graph forecasting. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4632–4650, Vienna, Austria, July 2025. Association for Computational Linguistics.

[Wang *et al.*, 2024a] Jiapu Wang, Sun Kai, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410, 2024.

[Wang *et al.*, 2024b] Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 8384–8410. Curran Associates, Inc., 2024.

[Xia *et al.*, 2024] Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. Chain-of-history reasoning for temporal knowledge graph forecasting. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16144–16159, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Xu *et al.*, 2020] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

[Xu *et al.*, 2023a] Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. Pre-trained language model with prompts for temporal knowledge graph completion. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7790–7803, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Xu *et al.*, 2023b] Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. Temporal knowledge graph reasoning with historical contrastive learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):4765–4773, Jun. 2023.

[Yang and Zhu, 2025] Xi Yang and Jia Zhu. Large language model with iteratively prompt for temporal knowledge graph completion. *Neurocomputing*, 651:130941, 2025.

[Zhong *et al.*, 2023] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4), November 2023.

[Zhu *et al.*, 2021] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4732–4740, May 2021.

[Zhu *et al.*, 2025] Peng Zhu, Yuante Li, Qinyuan Liu, Dawei Cheng, and Changjun Jiang. Financial time series prediction with multi-granularity graph augmented learning. *IEEE Transactions on Knowledge and Data Engineering*, 37(11):6436–6449, 2025.