From Implicit Graph Encoding to Explicit Evidence: A Training-Free LLM Framework for Temporal Knowledge Graph Reasoning

Guo Tang[1*], **Ke Cheng**[3*], **Huiming Fan**[1*], **Heng Chang**[4*‡], **Wenxiang Zheng**[1], **Xianhao Ou**[1], **Junjia Xiang**[1], **Ming Liu**[1,2†], **Yujun Zhou**[5], **Lanyu Li**[5], **Bing Qin**[1,2]
[1]Harbin Institute of Technology, Harbin, China [2]Peng Cheng Laboratory, Shenzhen, China
[3]Beihang University [4]Tsinghua University
[5]Nanjing Research Institute of Electronics Technology
{gtang,mliu}@ir.hit.edu.cn ckpassenger@buaa.edu.cn

Abstract

Temporal Knowledge Graph (TKG) forecasting faces significant challenges due to distribution shifts and poor inductive generalization in parametric models. While Large Language Models (LLMs) offer potent semantic reasoning, existing LLM-based approaches struggle with implicit modality alignment and suboptimal graph linearization, failing to capture complex topologies without retraining. To bridge this gap, we propose **ExE-LLM**, a training-free, test-time adaptive framework that reframes TKG prediction as explicit evidence-driven reasoning. Our core philosophy is to decouple topological calculation from semantic reasoning: a heuristic module translates latent graph signals into natural language evidence, enabling the LLM to perform multi-source judgment. ExE-LLM incorporates a task-aware scheduler for test-time adaptation, a heuristic synthesizer for explicit modality alignment, and a self-diagnosis module for iterative optimization. Extensive experiments on four benchmarks demonstrate that ExE-LLM achieves SOTA performance in inductive settings, significantly outperforming fully trained graph neural networks without updating LLM parameters. The source code is available at Github[1].

1 Introduction

Temporal Knowledge Graphs (TKGs) structurally model dynamic facts via timestamped triplets, serving as the backbone for recommendation systems and information retrieval (Leblay and Chekol, 2018; García-Durán et al., 2018; Xiang et al., 2022; Wang et al., 2019; Liu et al., 2018). Traditionally, dominant paradigms rely on parametric modeling, formulating link prediction as autoregressive generation or path-based reasoning over learned graph features (Li et al., 2022; Zhang et al., 2024; Li et al., 2021). While robust in closed environments, these methods exhibit a heavy reliance on training distributions: they often falter under distribution shifts, requiring retraining or fine-tuning, and struggle to generalize in inductive settings involving unseen entities or relations.

Recently, Large Language Models (LLMs) have been explored for TKG tasks due to their semantic reasoning capabilities (Yang et al., 2023; Yuan et al., 2024). As shown in Fig. 1(a), early approaches typically linearize graph structures into text (Lee et al., 2023; Xia et al., 2024), which often leads to information loss and context truncation, especially in long-horizon temporal reasoning (Chang et al., 2025). Subsequent research introduced soft prompts or graph adapters to align graph and text modalities (Jiang et al., 2025; Chang et al., 2025) (see Fig. 1(b)). Yet, two critical bottlenecks remain: (1) Superficial Modality Alignment: Hard-coded sequences or implicit projections fail to enable LLMs to genuinely perceive high-dimensional graph structures and temporal evolution patterns; (2) Residual Dependency: Most approaches still require co-training with GNN components, sacrificing the zero-shot capability of LLMs and limiting their utility in dynamic, open-ended environments.

To address these limitations, we propose a paradigm shift: reframing TKG link prediction from "*implicit topological inference via sequences*" to "*explicit evidence-driven semantic reasoning*". Our core idea is to decouple graph computation from reasoning (see Fig. 1(c)). Since LLMs excel at logical deduction but struggle with sparse graph signals, we design a training-free heuristic computation module acting as a lightweight feature extractor. This module explicitly translates graph signals—such as structural paths, temporal windows, and semantic correlations—into comprehensible natural language evidence for LLMs. Consequently, the inference task is transformed into

[*]Equal Contribution.
[†]Corresponding Author.
[‡]Project Leader.
[1]https://github.com/JENLISA4EVER/ExE-LLM

Findings of the Association for Computational Linguistics: ACL 2026, pages 5383–5405
July 2-7, 2026

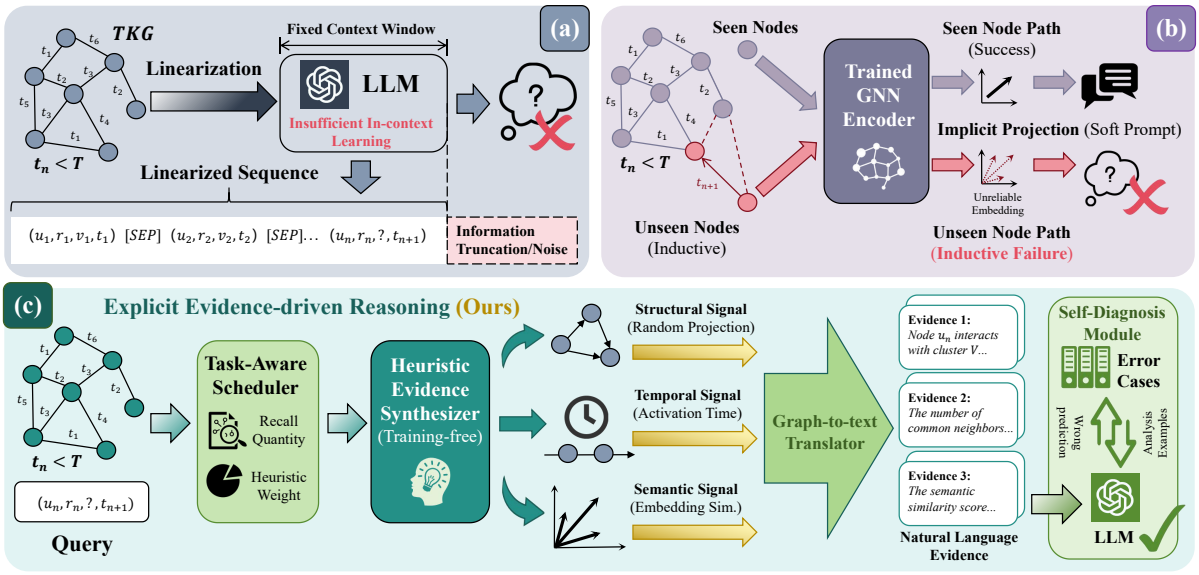


Figure 1: Comparison of LLM-based paradigms for TKG prediction. (a) Sequence linearization for in-context learning; (b) Soft prompt or adapter-based graph–text alignment; (c) Our paradigm decouples graph computation from LLM reasoning, enabling explicit evidence-driven and inductive inference.

a comprehensive reasoning process supported by multi-source heterogeneous evidence, preserving structural precision while leveraging LLM reasoning.

Based on this, we introduce **ExE-LLM** (**Ex**plicit **E**vidence for LLM Reasoning), a **training-free, test-time adaptive** framework. It comprises three core components: (1) *Task-Aware Scheduler*: Generates dynamic runtime parameters with query-aware context analysis, bypassing reliance on static training distributions. (2) *Heuristic Evidence Synthesizer*: Synthesizes graph signals into natural language evidence via multi-path retrieval and heuristic ranking, bridging the graph-text modality gap. (3) *Self-Diagnosis Module*: Enables immediate adaptation to inductive scenarios via feedback-driven failure analysis, eliminating the need for retraining.

We conducted a comprehensive evaluation on four standard benchmarks (ICEWS14/15/1819, GDELT) (García-Durán et al., 2018; Jin et al., 2019; Leetaru and Schrodt, 2013) covering link prediction and node retrieval tasks (Zhang et al., 2024). Using LLM as the core reasoner, **ExE-LLM achieves SOTA performance in inductive settings**, significantly outperforming baselines including trained GNNs. Ablation studies further verify the necessity of multi-path retrieval and dynamic scoring, as well as the interpretability of adaptive parameters.

Our contributions are summarized as follows:

- **A training-free test-time adaptive framework**: ExE-LLM achieves efficient adaptation to dynamic scenarios through test-time computation without gradient-based training, extending inductive generalization.
- **A novel paradigm for explicit modality alignment**: We move beyond implicit projection, utilizing heuristic computation to "translate" graph and temporal information into natural language evidence, achieving faithful alignment between graph topology and the LLM semantic space.
- **New SOTA in inductive settings**: Our approach significantly outperforms fully trained specialized graph models in inductive settings, validating the efficacy of "reasoning over memorization" in dynamic graphs.

2 Preliminaries

Temporal Knowledge Graph A Temporal Knowledge Graph (TKG) is defined as a temporally ordered sequence of facts, denoted by $\mathcal{G} = e_1, e_2, \ldots, e_N$. Each fact e_k is a quadruple (u, r, v, t), where $u, v \in \mathcal{E}$ are the head and tail entities from the entity set $\mathcal{E}$, $r \in \mathcal{R}$ is a relation from the relation set $\mathcal{R}$, and $t \in \mathcal{T}$ is the timestamp from a discrete timestamp set $\mathcal{T}$. The sequence follows chronological order: for any $k < j$, the timestamps satisfy $t_k \leq t_j$. Equivalently, a TKG can be represented as a sequence of timestamp-specific subgraphs $\mathcal{G} = \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{|\mathcal{T}|}$, where $\mathcal{G}_t$

contains all facts occurring at timestamp t.

Link Prediction on TKG Link prediction on TKG aims to forecast future facts based on historical observations. Given the historical graph $\mathcal{G}_{<t_q} = \{(u, r, v, t) \in \mathcal{G} | t < t_q\}$, which includes all facts before a query timestamp t_q, the task is to predict the missing tail entity for a query $(u_q, r_q, ?, t_q)$. The model ranks candidate entities $v \in \mathcal{E}$ according to their plausibility of completing the query quadruple, with the objective of assigning the highest rank to the ground-truth entity v_q.

3 Methodology

We present ExE-LLM, a training-free, test-time adaptive framework that decouples graph computation from semantic reasoning, addressing graph-text modality alignment while improving generalization via dynamic parameterization. As illustrated in Fig. 2, given a query, ExE-LLM derives adaptive inference hyperparameters from both local query context and global TKG statistics. Guided by these parameters, it retrieves and ranks candidate nodes from multiple graph perspectives using a bidirectional heuristic model, and synthesizes ranked candidates and explicit graph signals into structured textual evidence. Together with dynamically selected few-shot demonstrations, this evidence prompts the LLM for multi-evidence reasoning. Finally, ExE-LLM extracts reusable reasoning cases from erroneous predictions and maintains a dynamic case library to support subsequent inference.

3.1 Task-Aware Scheduler

The Task-Aware Scheduler aims to adaptively configure the inference strategy for each query by analyzing its specific structural and temporal context. It computes query-specific hyperparameters $\Theta = \{k_{str}, k_{tem}, k_{sem}, \alpha, \beta\}$ from local history of u and global TKG statistics, controlling recall budgets and heuristic fusion weights (details in App. A).

Query-Aware Context Analysis extracts three signals for node u at time T: structural popularity $\ell_u = \log(d_u + 1)$, temporal recency $\Delta T_u = T - t_{last}$, and semantic ambiguity σ_u (the standard deviation of similarity scores to top semantic neighbors). These signals characterize whether u is frequent, recent, or ambiguous.

Dynamic Parameter Generation uses the above signals to set recall budgets $k_{str}, k_{tem}, k_{sem}$ and fusion weights (α, β). Larger ℓ_u or smaller ΔT_u increases recall, while (α, β) trade off structural/semantic similarity and history-driven consistency.

3.2 Heuristic Evidence Synthesizer

The synthesizer is designed to bridge the modality gap by transforming raw graph signals into prioritized natural language evidence. It first retrieves potential candidate nodes through multiple graph paths, then evaluates their relevance using a bidirectional heuristic scoring mechanism, and finally translates the ranked candidates and graph signals related to q into structured textual descriptions for the LLM. We summarize the procedure in Algorithm 1.

Multi-path Recall constructs $\mathcal{C} = \mathcal{V}str \cup \mathcal{V}tem \cup \mathcal{V}sem$ with dynamically allocated budgets $(kstr, k_{tem}, k_{sem})$. $\mathcal{V}str$ contains nodes with the largest common-neighbor counts w.r.t. u; $\mathcal{V}tem$ contains the most recently activated nodes before T; $\mathcal{V}_{sem}$ contains nearest neighbors of u in the pretrained embedding space.

Heuristic Ranking Heuristic ranking follows the principle that a plausible future link should be simultaneously consistent with historical interaction patterns and homogeneous with the query node in both structural and semantic spaces. Given the candidate set $\mathcal{C}$, we assign each $v \in \mathcal{C}$ a scalar score by bidirectionally modeling interaction compatibility and feature similarity, and rank candidates accordingly.

Source-centric modeling. We first evaluate how well a candidate v aligns with the historical interaction context of the query node u. Let $\mathcal{H}_u = \{(p_i, r_i, t_i) \mid t_i < T\}$ denote u's historical interactions before time T. Each history item is weighted by an exponential time decay

$$w_i = \frac{\exp(-\lambda(T - t_i))}{\sum_j \exp(-\lambda(T - t_j))}. \quad (1)$$

For candidate v, we compute its similarity to each historical partner p_i in both structural and semantic spaces. When relation semantics are enabled, the semantic context is shifted by the corresponding relation embedding. The source-centric score is

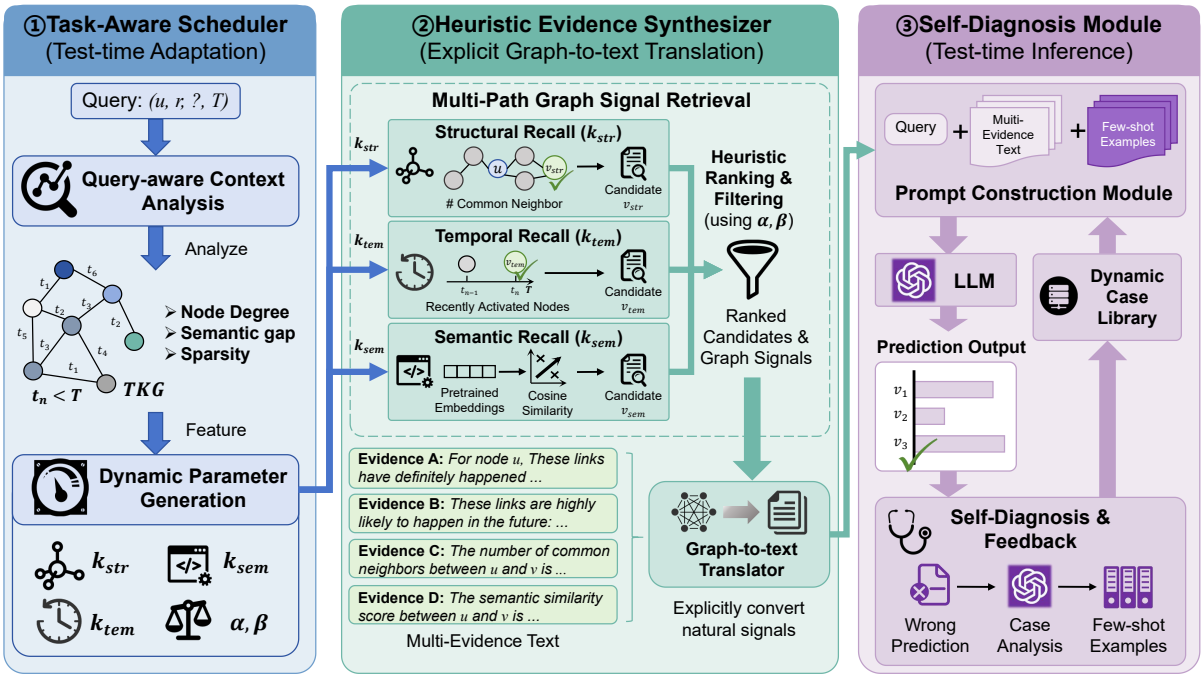


Figure 2: Overview of ExE-LLM. (1) Task-Aware Scheduler performs query-aware analysis of the query and global statistics to generate adaptive runtime parameters. (2) Heuristic Evidence Synthesizer retrieves multi-view candidates and translates latent graph signals into explicit natural language evidence via bidirectional heuristic ranking. (3) Self-Diagnosis Module optimizes the inference context by analyzing failure cases to construct dynamic few-shot demonstrations.

defined as

$$s_{\text{src}}(v) = \sum_i w_i \Big[\alpha \cos(\mathbf{z}_v^{\text{str}}, \mathbf{z}_{p_i}^{\text{str}}) + (1-\alpha)\cos(\mathbf{z}_v^{\text{sem}}, \mathbf{z}_{p_i}^{\text{sem}} + \mathbf{z}_{r_i}^{\text{rel}})\Big], \tag{2}$$

where $\mathbf{z}^{\text{str}}$ and $\mathbf{z}^{\text{sem}}$ denote structural and semantic embeddings, respectively, and α balances the two spaces.

Destination-centric modeling. To capture reciprocal compatibility, we further assess whether the query node u is consistent with the historical neighborhood of candidate v. Let $\mathcal{H}_v = \{(p_j, t_j) \mid t_j < T\}$ be v's valid history, and define time-decayed weights $\tilde{w}_j$ analogously. We aggregate v's history into prototype embeddings

$$\bar{\mathbf{z}}_v^{\text{sem}} = \sum_j \tilde{w}_j \mathbf{z}_{p_j}^{\text{sem}}, \bar{\mathbf{z}}_v^{\text{str}} = \sum_j \tilde{w}_j \mathbf{z}_{p_j}^{\text{str}}, \tag{3}$$

and compute

$$s_{\text{str}}(v) = \cos(\mathbf{z}_u^{\text{str}}, \bar{\mathbf{z}}_v^{\text{str}}), \tag{4}$$
$$s_{\text{sem}}(v) = \cos(\mathbf{z}_u^{\text{sem}}, \bar{\mathbf{z}}_v^{\text{sem}}), \tag{5}$$
$$s_{\text{dst}}(v) = \alpha s_{\text{str}}(v) + (1-\alpha) s_{\text{sem}}(v). \tag{6}$$

Self-similarity. Independently, we measure direct homophily between u and v:

$$s_{\text{str}}^{(\text{self})}(v) = \cos(\mathbf{z}_u^{\text{str}}, \mathbf{z}_v^{\text{str}}), \tag{7}$$
$$s_{\text{sem}}^{(\text{self})}(v) = \cos(\mathbf{z}_u^{\text{sem}}, \mathbf{z}_v^{\text{sem}}), \tag{8}$$
$$s_{\text{self}}(v) = \alpha s_{\text{str}}^{(\text{self})}(v) + (1-\alpha) s_{\text{sem}}^{(\text{self})}(v). \tag{9}$$

Score fusion and ranking. The interaction consistency score is obtained by averaging the two directional terms,

$$s_{\text{int}}(v) = \begin{cases} \frac{s_{\text{src}}(v) + s_{\text{dst}}(v)}{2}, & \text{if bilateral modeling,} \\ s_{\text{src}}(v), & \text{otherwise.} \end{cases} \tag{10}$$

and the final heuristic score is

$$S(v) = \beta s_{\text{int}}(v) + (1-\beta) s_{\text{self}}(v), \tag{11}$$

where β controls the reliance on historical interaction patterns versus intrinsic similarity. Candidates are ranked by $S(v)$ in descending order, yielding a prioritized list that balances temporal consistency, structural proximity, and semantic alignment for downstream evidence synthesis.

Graph-to-Text Translator After ranking, we partition the scored candidate set $\mathcal{C}$ into potential future, ambiguous, and unlikely subsets, and verbalize them as category-wise evidence. Final decisions are made pointwise over an option set $\mathcal{O}$: for each (u, r, o, T), we translate explicit signals between u and o (e.g., common neighbors, past frequency, semantic similarity) into an *Additional Context* block.

3.3 Self-Diagnosis Module

This module implements a lightweight inference-time feedback mechanism without parameter updates. For each incorrect pointwise prediction (u, r, o, T), ExE-LLM reconstructs the original multi-evidence prompt and queries the LLM for a brief explanation conditioned on the ground-truth outcome. The resulting explanation, together with the corresponding context and correct decision, is stored as a reasoning case in a bounded dynamic case library.

During streaming inference, ExE-LLM incrementally updates (i) structural representations via random-projection-based aggregation, (ii) temporal statistics such as recent node activation times, and (iii) the few-shot case library based on observed failures. A small number of recent cases are injected as few-shot demonstrations for subsequent queries, providing adaptive reasoning priors aligned with evolving graph dynamics. Prompt templates are shown in Fig. 9 and 10. App. E provides details of the LLM scoring functions.

4 Experimental Setup

Datasets We conduct experiments on two widely-used TKG datasets: the Integrated Crisis Early Warning System (ICEWS) dataset, including ICEWS14 (García-Durán et al., 2018), ICEWS05-15 (García-Durán et al., 2018), and ICEWS1819 (Jin et al., 2019) versions, and the Global Database of Events, Language, and Tone (GDELT) dataset (Leetaru and Schrodt, 2013) (detailed in Appendix B). Table 7 presents the statistics of the datasets.

Baselines For dynamic graph models, we use standard baselines from the DTGB benchmark[2] (Zhang et al., 2024), including JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), CAWN (Wang et al., 2022), TCL (Wang et al., 2021), GraphMixer (Cong et al., 2023), and DyGFormer (Yu et al., 2023). Regarding specialized TKG forecasting models, we include strong baselines such as RE-GCN (Li et al., 2021), TiRGN (Li et al., 2022), and DiffuTKG (Cai et al., 2024). Finally, for training-free LLM-based approaches, we compare against ICL (Lee et al., 2023), GAD (Lei et al., 2025), and AnRe (Tang et al., 2025). Detailed descriptions of these baselines are provided in Appendix C.

Tasks & Evaluation Metrics We conduct experiments on two fundamental tasks within both transductive and inductive settings. Inductive setting follows DTGB: nodes appearing in test are unseen during training. For the future link prediction task, we report the standard metrics: Area Under the ROC Curve (AUC-ROC) and Average Precision (AP) scores. For the node retrieval task, we present the Hits@K metric, assessing performance at K=1, 3, and 10.

Implementation Details Implemented in PyTorch, our framework utilizes Qwen3-8B[3] (Yang et al., 2025) as the backbone and all-mpnet-base-v2[4] for semantic encoding. Structural features are derived via the Random Projection Module (Lu et al., 2024) (see Appendix D). Experiments are conducted on an NVIDIA A100 (80GB) GPU. Following standard protocols (Zhang et al., 2024), datasets are split chronologically (70/15/15%). Results report the *mean* ± *std* over three independent runs. Additional implementation details are provided in Appendix F.

5 Experimental Results

5.1 Main Results

Link Prediction Table 1 reports link prediction results under *global random* negatives, with complete results in Table 8. Dynamic-graph baselines are taken from DTGB (Zhang et al., 2024). We summarize the key observations below:

- Under the *transductive* setting, LLM-based reasoning methods are constrained by limited access to structural and temporal signals, leading to lower performance than trained GNN-based approaches. Dynamic graph models that jointly capture temporal and structural information achieve the strongest results. By explicitly integrating

[2] https://github.com/zjs123/DTGB

[3] https://huggingface.co/Qwen/Qwen3-8B

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Setting	Model	ICEWS14		ICEWS05-15		ICEWS1819		GDELT	
		AP	AUC	AP	AUC	AP	AUC	AP	AUC
tr.	**TGAT**	97.65 ± 0.08	97.42 ± 0.10	98.94 ± 0.03	98.85 ± 0.02	98.05 ± 0.52	97.87 ± 0.65	93.42 ± 0.39	93.41 ± 0.46
	CAWN	96.79 ± 0.14	96.12 ± 0.17	98.45 ± 0.08	98.16 ± 0.10	98.38 ± 0.31	98.15 ± 0.41	93.98 ± 0.36	94.19 ± 0.26
	DyGFormer	97.77 ± 0.06	97.34 ± 0.07	99.08 ± 0.02	98.93 ± 0.03	98.84 ± 0.13	98.65 ± 0.24	96.40 ± 0.02	96.48 ± 0.07
	RE-GCN	96.26 ± 0.14	95.99 ± 0.15	97.91 ± 0.06	97.77 ± 0.06	98.39 ± 0.25	98.35 ± 0.24	95.66 ± 0.08	95.72 ± 0.07
	TiRGN	96.20 ± 0.08	95.99 ± 0.07	97.45 ± 0.14	97.27 ± 0.15	97.85 ± 0.20	97.84 ± 0.18	94.80 ± 0.52	94.87 ± 0.59
	DiffuTKG	97.11 ± 0.02	96.90 ± 0.02	98.70 ± 0.03	98.57 ± 0.03	98.77 ± 0.07	98.71 ± 0.07	94.80 ± 0.10	95.52 ± 0.03
	ICL	86.11 ± 0.05	85.47 ± 0.02	90.10 ± 0.01	90.08 ± 0.03	89.60 ± 0.04	88.63 ± 0.04	90.56 ± 0.06	90.33 ± 0.07
	GAD	85.60 ± 1.01	83.34 ± 1.05	86.77 ± 0.87	84.96 ± 0.74	84.35 ± 1.03	82.48 ± 1.03	85.56 ± 0.64	84.71 ± 0.74
	AnRe	86.45 ± 0.04	86.34 ± 0.02	91.88 ± 0.07	91.76 ± 0.08	90.98 ± 0.09	90.00 ± 0.08	83.71 ± 0.02	81.86 ± 0.01
	ExE-LLM	**98.01 ± 0.04**	**97.94 ± 0.03**	**99.15 ± 0.05**	**99.02 ± 0.06**	**99.18 ± 0.03**	**99.15 ± 0.03**	**97.69 ± 0.08**	**97.73 ± 0.09**
in.	**TGAT**	92.92 ± 0.12	92.64 ± 0.11	96.58 ± 0.08	96.24 ± 0.09	91.81 ± 0.56	91.51 ± 0.61	75.00 ± 0.67	75.01 ± 0.74
	CAWN	90.50 ± 0.41	88.50 ± 0.43	95.51 ± 0.24	94.44 ± 0.28	94.06 ± 0.28	93.30 ± 0.76	79.80 ± 0.16	79.09 ± 0.10
	DyGFormer	92.84 ± 0.13	91.35 ± 0.12	97.39 ± 0.05	96.85 ± 0.04	96.65 ± 0.11	96.13 ± 0.10	91.72 ± 0.14	91.35 ± 0.24
	RE-GCN	88.60 ± 0.34	88.20 ± 0.41	94.00 ± 0.13	93.80 ± 0.13	94.84 ± 0.75	94.78 ± 0.73	82.69 ± 0.43	82.19 ± 0.36
	TiRGN	89.09 ± 0.30	88.95 ± 0.36	92.58 ± 0.43	92.32 ± 0.43	93.95 ± 0.54	94.10 ± 0.45	80.92 ± 1.46	80.22 ± 1.53
	DiffuTKG	90.07 ± 0.32	89.42 ± 0.34	96.08 ± 0.03	95.66 ± 0.05	96.04 ± 0.17	95.80 ± 0.19	83.60 ± 0.08	84.96 ± 0.06
	ICL	86.04 ± 0.01	85.44 ± 0.03	89.97 ± 0.02	89.99 ± 0.05	88.33 ± 0.06	87.26 ± 0.05	90.02 ± 0.01	89.76 ± 0.01
	GAD	84.77 ± 1.44	82.78 ± 1.25	85.23 ± 0.70	83.32 ± 0.65	83.50 ± 0.02	81.59 ± 0.02	84.94 ± 2.25	84.12 ± 2.40
	AnRe	85.12 ± 0.02	84.86 ± 0.03	90.97 ± 0.03	90.75 ± 0.06	90.47 ± 0.16	89.36 ± 0.19	80.78 ± 0.12	78.43 ± 0.05
	ExE-LLM	**97.99 ± 0.03**	**97.91 ± 0.05**	**99.06 ± 0.02**	**98.96 ± 0.03**	**99.14 ± 0.06**	**99.13 ± 0.05**	**98.85 ± 0.04**	**98.83 ± 0.02**

Table 1: Under the global negative sampling strategy, the performance comparison of various models on the link prediction task. *tr.* means transductive setting, and *in.* means inductive setting. The results of the dynamic graph models, TKG-specific models, and training-free LLM methods are respectively presented in the white, gray, and blue tables. The best performance within each setting is highlighted in **bold**.

textual, structural, and temporal evidence, ExE-LLM attains state-of-the-art performance.

- Under the *inductive* setting, trained GNN-based methods exhibit notable performance degradation, while LLM-based reasoning methods remain largely stable, likely because they do not rely on learned entity embeddings. This highlights the advantage of LLM-based reasoning for zero-shot dynamic adaptation. Benefiting from dynamic parameter coordination and self-diagnosis, ExE-LLM significantly outperforms all baselines.

Node Retrieval To evaluate robustness under different negative samples, we adopt DTGB's global random negative sampling to expand the candidate set to 99 negatives and rank 100 candidates (including the positive) by LLM prediction scores. As shown in Table 2, dynamic graph representation learning methods exhibit strong stability on transductive data, whereas LLM-based methods show lower stability due to limited feature interaction and high variance over multiple-choice outputs. Despite this, ExE-LLM outperforms most trained GNN-based methods and substantially surpasses other LLM-based approaches. Under the inductive setting, trained GNN methods degrade significantly, while ExE-LLM maintains strong performance, highlighting its superior out-of-domain generalization. Complete results are reported in Table 9.

5.2 Ablation Study

As shown in Table 3, we conduct ablation studies on ICEWS1819 and GDELT to isolate the contribution of each component in ExE-LLM.

Task-Aware Scheduler: Fixing recall budgets (**w/o Adapt**) leads to consistent performance drops, confirming the necessity of query-time adaptation under temporal distribution shifts. Similarly, enforcing $\alpha=0$ (semantic-only) or $\alpha=1$ (structural-only) degrades accuracy, indicating that effective reasoning requires fusing both semantic and structural views. Moreover, setting $\beta=0$ (no history) or $\beta=1$ (no self-similarity) further reduces performance, verifying the complementarity between historical aggregation and direct affinity. **Multi-path Evidence Retrieval:** Relying on any single retrieval path (*time-active*, *structural*, or *semantic*) substantially degrades performance, demonstrating that recency, topology, and content provide complementary and non-redundant evidence. **Self-Diagnosis:** Removing dynamic few-shot learning (**w/o few-shot**) results in notable performance declines across datasets, highlighting the effectiveness of transforming model errors into reusable corrective exemplars.

Summary. Overall, the dynamic scheduler, multi-

Setting	Model	ICEWS14			ICEWS05-15			GDELT		
		Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
tr.	**TGAT**	63.29 ± 0.28	81.00 ± 0.35	**91.98 ± 0.32**	77.24 ± 0.32	**90.55 ± 0.64**	**96.49 ± 0.41**	41.35 ± 0.13	66.21 ± 0.19	88.17 ± 0.35
	CAWN	47.86 ± 2.63	53.40 ± 3.39	56.53 ± 3.57	58.21 ± 4.94	62.38 ± 5.53	64.88 ± 5.73	42.53 ± 0.10	66.31 ± 0.17	87.47 ± 0.74
	DyGFormer	69.61 ± 0.11	81.26 ± 0.18	86.12 ± 0.22	**78.01 ± 0.32**	86.90 ± 0.24	91.30 ± 0.23	**47.64 ± 0.08**	**70.89 ± 0.10**	**90.16 ± 0.11**
	RE-GCN	40.61 ± 0.26	73.33 ± 0.21	86.66 ± 1.00	66.96 ± 0.29	81.26 ± 0.90	89.31 ± 1.45	OOM	OOM	OOM
	TiRGN	45.60 ± 0.83	77.29 ± 1.74	85.23 ± 1.95	64.97 ± 1.03	84.97 ± 1.03	89.95 ± 0.54	OOM	OOM	OOM
	DiffuTKG	53.07 ± 2.03	78.29 ± 2.04	87.48 ± 2.06	70.39 ± 0.95	86.63 ± 0.95	91.85 ± 0.94	OOM	OOM	OOM
	ICL	30.77 ± 0.07	35.77 ± 0.04	43.22 ± 0.14	36.05 ± 0.05	47.09 ± 0.09	50.02 ± 0.13	16.91 ± 0.03	21.01 ± 0.37	29.09 ± 0.45
	GAD	40.45 ± 0.41	45.10 ± 0.32	49.88 ± 0.52	43.10 ± 0.98	49.37 ± 0.45	54.87 ± 0.08	27.95 ± 0.48	31.50 ± 0.43	36.71 ± 0.64
	AnRe	59.26 ± 0.09	63.11 ± 0.06	67.09 ± 0.09	64.77 ± 0.75	71.18 ± 0.04	75.12 ± 0.32	38.18 ± 0.17	50.11 ± 0.18	52.51 ± 0.91
	ExE-LLM	**71.80 ± 0.02**	**82.63 ± 0.05**	88.50 ± 0.09	77.93 ± 0.04	86.93 ± 0.06	91.83 ± 0.05	44.10 ± 0.14	65.77 ± 0.21	84.97 ± 0.11
in.	**TGAT**	37.35 ± 0.33	57.10 ± 0.42	77.47 ± 0.38	56.64 ± 0.41	76.64 ± 0.63	89.83 ± 0.54	26.30 ± 0.42	48.18 ± 0.55	73.29 ± 0.29
	CAWN	36.30 ± 1.94	44.49 ± 2.86	49.48 ± 3.27	49.29 ± 3.97	55.59 ± 5.11	59.25 ± 5.36	29.16 ± 0.11	46.67 ± 0.61	69.25 ± 0.19
	DyGFormer	50.18 ± 0.34	65.11 ± 0.65	74.60 ± 0.32	63.94 ± 0.45	78.32 ± 0.32	86.12 ± 0.24	36.81 ± 0.38	56.66 ± 0.26	78.44 ± 0.18
	RE-GCN	34.64 ± 0.21	54.26 ± 0.11	65.41 ± 1.25	46.78 ± 1.22	66.32 ± 1.08	76.13 ± 2.20	OOM	OOM	OOM
	TiRGN	30.25 ± 0.92	58.25 ± 1.44	72.21 ± 1.98	49.83 ± 0.58	70.84 ± 0.28	83.64 ± 1.38	OOM	OOM	OOM
	DiffuTKG	48.87 ± 2.97	69.20 ± 2.95	79.44 ± 2.94	57.73 ± 1.41	75.04 ± 1.40	85.34 ± 1.41	OOM	OOM	OOM
	ICL	28.24 ± 0.06	33.67 ± 0.03	42.07 ± 0.08	35.14 ± 0.34	43.55 ± 0.08	49.97 ± 0.03	15.48 ± 0.34	19.77 ± 0.41	26.51 ± 0.45
	GAD	37.02 ± 0.25	44.12 ± 0.90	48.42 ± 0.29	43.01 ± 0.76	46.62 ± 0.73	54.00 ± 0.58	19.65 ± 0.56	23.41 ± 0.59	29.45 ± 0.68
	AnRe	54.84 ± 0.11	61.08 ± 0.43	66.63 ± 0.67	62.36 ± 0.12	70.94 ± 0.08	73.43 ± 0.09	35.58 ± 0.78	45.50 ± 0.85	48.60 ± 0.88
	ExE-LLM	**68.10 ± 0.04**	**78.77 ± 0.07**	**86.17 ± 0.02**	**74.87 ± 0.02**	**84.83 ± 0.06**	**90.27 ± 0.10**	**42.93 ± 0.13**	**63.70 ± 0.17**	**82.63 ± 0.22**

Table 2: Under the global negative sampling strategy, the performance comparison of various models on the node retrieval task. *tr.* means transductive setting, and *in.* means inductive setting. OOM means out-of-memory. The results of the dynamic graph models, TKG-specific models, and training-free LLM methods are respectively presented in the white, gray, and blue tables. The best performance within each setting is highlighted in **bold**.

Variant	ICEWS1819		GDELT	
	AP	AUC	H@1	H@10
tem-recall	98.27	98.20	28.67	67.98
struc-recall	97.97	97.91	39.06	79.42
sem-recall	98.54	98.49	43.15	83.93
w/o adapt	98.61	98.56	38.93	79.28
w/o struc-sim	98.04	97.96	36.95	77.10
w/o sem-sim	98.57	98.52	37.23	77.84
w/o int-score	97.49	97.43	30.99	70.53
w/o self-score	98.46	98.41	38.14	78.41
w/o few-shot	98.61	98.56	38.63	78.95
ExE-LLM	**99.18**	**99.15**	**44.10**	**84.97**

Table 3: Ablation study of ExE-LLM. ICEWS1819 reports AP/AUC. GDELT reports Hits@1/Hits@10.

path evidence retrieval, and self-diagnosis module are all indispensable for robust reasoning on streaming temporal knowledge graphs, jointly enabling ExE-LLM to maintain high predictive accuracy in dynamic environments.

6 Analysis and Discussion

6.1 Hard Negative

To evaluate ExE-LLM's ability to rank hard negative samples retrieved by heuristic recall strategies, we construct the negative sampling pool exclusively from samples obtained via three recall strategies and test all LLM-based reasoning methods under this setting. As shown in Table 4, the experiments reveal that when standard negative samples are replaced with hard negatives, the performance of all methods drops noticeably. Nevertheless, ExE-LLM maintains a relatively high accuracy, demonstrating that part of its effectiveness stems from the LLM's enhanced capacity to leverage and reason over multiple sources of evidence.

6.2 Parameter Sensitivity

Figure 4 reports recall curves under three strategies as the recall budget (k) varies, while Figure 3 shows the frequency distributions of (α) and (β). Across both ICEWS1819 and GDELT, the structural budget (k_{struc}) exhibits the largest variation, whereas (k_{time}) remains stable and (k_{sem}) varies only mildly. This pattern suggests that structural dynamics—such as degree bursts and community rewiring—constitute the primary source of non-stationarity, while event rates are relatively stable and semantic embeddings remain well calibrated. Accordingly, the scheduler expands (k_{struc}) to accommodate structural shifts, while keeping a consistent temporal window and adjusting semantic recall when necessary. The $((\alpha, \beta))$ heatmaps peak around $(\alpha \approx 0.73\text{–}0.75)$ and $(\beta \approx 0.65)$, indicating a preference for structural similarity and historical aggregation. Emphasizing structural signals helps filter semantically plausible but structurally implausible distractors, while a moderately high (β) stabilizes decisions by aggregating reliable historical partners without suppressing emerging nodes. Overall, these adaptive behaviors align

Model	ICEWS1819				GDELT			
	tr.		*in.*		*tr.*		*in.*	
	AP	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC
ICL	86.48 ± 0.12	85.51 ± 0.07	85.32 ± 0.09	84.87 ± 0.06	74.62 ± 0.07	74.07 ± 0.06	74.58 ± 0.08	74.02 ± 0.05
GAD	83.53 ± 0.72	82.35 ± 0.81	83.41 ± 0.65	82.28 ± 0.75	71.45 ± 0.16	70.30 ± 0.11	71.38 ± 0.14	70.25 ± 0.09
AnRe	91.47 ± 0.34	91.26 ± 0.31	91.39 ± 0.28	91.19 ± 0.28	80.24 ± 0.02	79.90 ± 0.21	81.19 ± 0.04	80.15 ± 0.18
ExE-LLM	**94.19 ± 0.06**	**94.10 ± 0.05**	**94.14 ± 0.05**	**94.06 ± 0.04**	**87.24 ± 0.08**	**87.12 ± 0.04**	**87.97 ± 0.07**	**87.88 ± 0.03**

Table 4: Under the recall-pool negative sampling strategy, the performance comparison of LLM-based models on the link prediction task. *tr.* means transductive setting, and *in.* means inductive setting. The best performance is highlighted in **bold**.

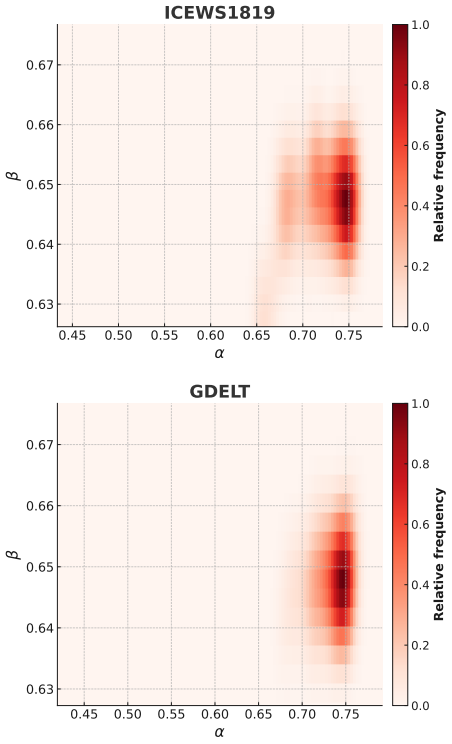


Figure 3: The relative frequency distribution of (α, β) on ICEWS1819 and GDELT.

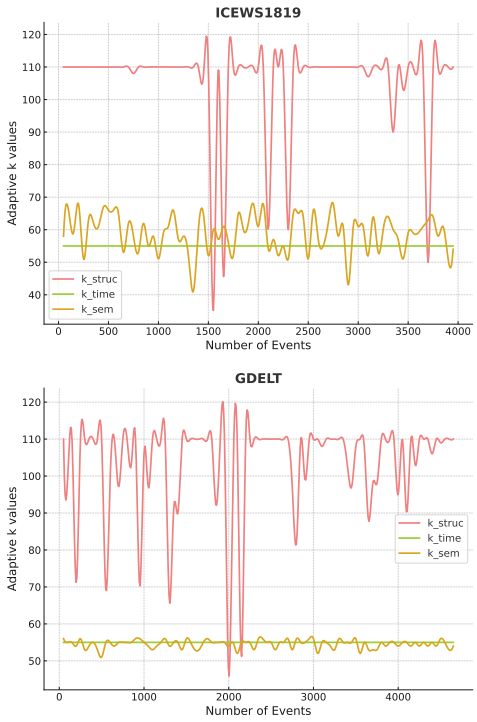


Figure 4: Under ICEWS1819 and GDELT, the adaptive changes of three recall quantities k as the test progresses.

with the underlying data characteristics, reflecting structure-dominated dynamics alongside steady temporal and semantic patterns. Appendix G provides a hyperparameter sensitivity analysis of ExE-LLM.

6.3 Contribution of the LLM

To quantify the impact of heuristic computation and LLMs on prediction, we conducted an analysis by removing the LLM and predicting results based solely on the heuristic scores (see Tab. 5-6). The results show a drastic performance drop without LLM reasoning. In our framework, the heuristic ranker serves as a "coarse-grained" filter to narrow down the search space. The final "fine-grained" prediction relies on the LLM's ability to synthesize and reason over textual evidence, demonstrating that the LLM is essential for handling complex semantic entanglements and multi-source evidence.

6.4 Training and Inference Efficiency

We provide a runtime comparison between GNN models and LLM-based methods in Figure 5. Training-free LLM methods eliminate the substantial resource consumption required for offline multi-epoch training (which can take dozens of hours on large datasets). Furthermore, ExE-LLM avoids the heavy neighborhood sampling overhead required by GNNs during inference. While slightly slower than other prompting methods, ExE-LLM provides a superior balance between accuracy and runtime

Setting	Variant	ICEWS14		ICEWS05-15		ICEWS1819		GDELT	
		AP	*AUC*	*AP*	*AUC*	*AP*	*AUC*	*AP*	*AUC*
tr.	w/o LLM	61.42	60.15	65.03	64.16	60.78	59.97	58.76	58.61
	ExE-LLM	**98.01**	**97.94**	**99.15**	**99.02**	**99.18**	**99.15**	**97.69**	**97.73**
in.	w/o LLM	59.76	59.85	64.12	63.30	60.54	60.20	59.77	59.04
	ExE-LLM	**97.99**	**97.91**	**99.06**	**98.96**	**99.14**	**99.13**	**98.85**	**98.83**

Table 5: Performance comparison on Link Prediction after removing LLM reasoning.

Setting	Variant	ICEWS14		ICEWS05-15		ICEWS1819		GDELT	
		Hit@1	*Hit@10*	*Hit@1*	*Hit@10*	*Hit@1*	*Hit@10*	*Hit@1*	*Hit@10*
tr.	w/o LLM	15.78	34.33	16.12	32.26	12.88	30.74	9.75	21.55
	ExE-LLM	**71.80**	**88.50**	**77.93**	**91.83**	**75.93**	**93.00**	**44.10**	**84.97**
in.	w/o LLM	15.56	33.89	15.75	31.78	13.03	30.11	9.57	20.67
	ExE-LLM	**68.10**	**86.17**	**74.87**	**90.27**	**74.00**	**91.73**	**42.93**	**82.63**

Table 6: Performance comparison on Node Retrieval after removing LLM reasoning.

efficiency for dynamic streaming applications.

7 Related Work

LLMs for temporal knowledge graphs. Recent studies explore large language models (LLMs) for temporal knowledge graph (TKG) forecasting by leveraging their semantic and reasoning capabilities. Early works (Peters et al., 2019; Han et al., 2023; Yang et al., 2024; Xu et al., 2024) linearize historical quadruples and repurpose pretrained LMs as temporal encoders. Subsequent methods integrate temporal order and structure via time-aware encodings and prompt designs (Jiang et al., 2023; Tan et al., 2023; Yuan et al., 2023). Parameter-efficient adaptations are explored by zr-LLM (Ding et al., 2024) and LLM-DA (Ye et al., 2024), while recent approaches favor prompting and generation-based paradigms, including autoregressive prediction with ICL (Shi et al., 2023; Lee et al., 2023; Zhang et al., 2025), few-shot instruction tuning (Liao et al., 2024), online co-evolution (Yu et al., 2024), and higher-order history abstraction (Xia et al., 2024). Despite these advances, most methods focus on quadruple-level prediction under coarse temporal granularity, leaving text-rich and fine-grained dynamic settings largely unexplored.

Dynamic graph neural networks. Dynamic graph learning without textual modeling follows two main paradigms. Snapshot-based methods (Pareja et al., 2019; Sankar et al., 2019; Xiao et al., 2025) discretize graphs over time, with DySAT (Sankar et al., 2019) jointly modeling structural and temporal dependencies via self-attention. Continuous-time approaches (Kumar et al., 2019; Trivedi et al., 2018; Zheng et al., 2025) model interactions as temporal point processes, exemplified by TGAT (Xu et al., 2020) and TGN (Rossi et al., 2020), while DyGLib (Yu et al., 2023) standardizes evaluation protocols. These methods rely on fixed parameters at inference time and cannot incorporate evolving textual semantics, leading to performance degradation under distribution shifts. Our method addresses this limitation via a training-free, test-time adaptive framework that jointly leverages temporal, structural, and textual signals.

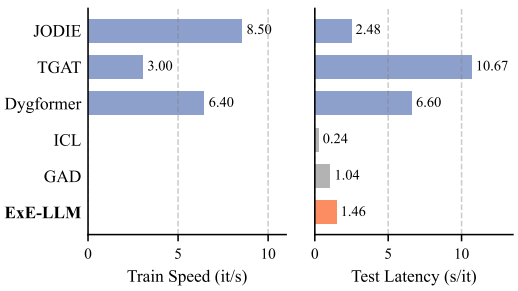


Figure 5: Efficiency comparison across models.

8 Conclusion

We propose ExE-LLM, a training-free and test-time adaptive framework for TKG reasoning. By decoupling graph computation from LLM-based semantic reasoning, ExE-LLM formulates TKG prediction as an explicit evidence-driven inference problem, enabling robust generalization without fine-tuning. Experiments on multiple benchmarks demonstrate that ExE-LLM consistently outperforms trained graph models and existing LLM-based methods, especially in inductive settings. These results suggest that explicit graph-to-text translation and test-time adaptation provide an effective and scalable paradigm for reasoning over evolving knowledge graphs.

Limitations

As discussed in our experimental analysis, while LLMs can achieve strong performance on classification tasks, they tend to underperform compared to traditional trained methods on ranking tasks due to high output variance—a common limitation of LLM-based approaches. We explored several strategies to mitigate this variance, such as averaging predictions over multiple runs and using larger-parameter models for inference; however, experiments showed limited effectiveness. We hypothesize that this issue stems from the inherent lim-

itations of non-finetuned LLMs, and future work may address it by introducing lightweight trainable components—e.g., a post-hoc classifier head.

Additionally, our current method relies on heuristic features for computation, yet certain modules are unnecessary in scenarios where distinctions are straightforward. Applying the same computational pipeline uniformly across all cases likely leads to inefficient resource usage. To address this, we plan to incorporate gating mechanisms in future work to dynamically activate only the necessary components based on input complexity.

Ethics Statement

All experiments utilized publicly available or synthetically generated datasets, which underwent anonymization procedures. We thoroughly assessed potential model biases and societal implications, while also evaluating the safety of generated content. The authors declare no conflicts of interest. The code and data associated with this study have been open-sourced to enhance transparency and reproducibility.

This article employed LLMs to refine certain aspects of writing logic and grammatical accuracy. In the experimental code section, some portions of the code were generated with the assistance of LLMs. However, LLMs were not involved in the formulation of the core ideas or the overall structure of the manuscript.

Acknowledgements

The research in this article is supported by the National Science Foundation of China (U22B2059, 62276083), Key Research and Development Program of Heilongjiang Providence (2022ZX01A28).

References

Yuxiang Cai, Qiao Liu, Yanglei Gan, Changlin Li, Xueyi Liu, Run Lin, Da Luo, and JiayeYang JiayeYang. 2024. Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5766–5778, Bangkok, Thailand. Association for Computational Linguistics.

He Chang, Jie Wu, Zhulin Tao, Yunshan Ma, Xianglin Huang, and Tat-Seng Chua. 2025. Integrate temporal graph learning into llm-based temporal knowledge graph model. *arXiv preprint arXiv:2501.11911*.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. 2023. Do we really need complicated model architectures for temporal networks? *Preprint*, arXiv:2302.11636.

Zifeng Ding, Heling Cai, Jingpei Wu, Yunpu Ma, Ruotong Liao, Bo Xiong, and Volker Tresp. 2024. zr-llm: Zero-shot relational learning on temporal knowledge graphs with large language models. *Preprint*, arXiv:2311.10112.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.

Zhen Han, Ruotong Liao, Beiyan Liu, Yao Zhang, Zifeng Ding, Jindong Gu, Heinz Koeppl, Hinrich Schuetze, and Volker Tresp. 2023. Enhanced temporal knowledge embeddings with contextualized language representations.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *Preprint*, arXiv:2305.09645.

Zihao Jiang, Ben Liu, Miao Peng, Wenjie Xu, Yao Xiao, Zhenyan Shan, and Min Peng. 2025. Towards explainable temporal reasoning in large language models: A structure-aware generative framework. *arXiv preprint arXiv:2505.15245*.

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

Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1269–1278. ACM.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. *arXiv preprint arXiv:2305.10613*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Runlin Lei, Jiarui Ji, Haipeng Ding, Lu Yi, Zhewei Wei, Yongchao Liu, and Chuntao Hong. 2025. Exploring the potential of large language models as predictors in dynamic text-attributed graphs. *Preprint*, arXiv:2503.03258.

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

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. 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. Association for Computing Machinery.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024. Gentkg: Generative forecasting on temporal knowledge graph with large language models. *Preprint*, arXiv:2310.07793.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *Cornell University - arXiv,Cornell University - arXiv*.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. 2024. Improving temporal link prediction via temporal walk matrix projection. *Preprint*, arXiv:2410.04013.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. 2019. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *Preprint*, arXiv:1902.10191.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. *Preprint*, arXiv:1909.04164.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *Preprint*, arXiv:2006.10637.

Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2019. Dynamic graph representation learning via self-attention networks. *Preprint*, arXiv:1812.09430.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023. Language models can improve event prediction by few-shot abductive reasoning. *Preprint*, arXiv:2305.16646.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. *Preprint*, arXiv:2306.08952.

Guo Tang, Zheng Chu, Wenxiang Zheng, Junjia Xiang, Yizhuo Li, Weihao Zhang, Ming Liu, and Bing Qin. 2025. AnRe: Analogical replay for temporal knowledge graph forecasting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4632–4650, Vienna, Austria. Association for Computational Linguistics.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2018. Representation learning over dynamic graphs. *Preprint*, arXiv:1803.04051.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. Dyrep: Learning representations over dynamic graphs. *International Conference on Learning Representations (ICLR)*.

Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. 2021. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *Preprint*, arXiv:2105.07944.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. 2022. Inductive representation learning in temporal networks via causal anonymous walks. *Preprint*, arXiv:2101.05974.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16144–16159, Bangkok, Thailand. Association for Computational Linguistics.

Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. 2022. Temporal and heterogeneous graph neural network for financial time series prediction.

Peng Xiao, Chao Liu, Wei Jia, and Lijun Dong. 2025. Aligned-entities-based fusion embedding on hetero-field knowledge graphs. *DATA INTELLIGENCE*, 7(3):618–635.

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

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2024. Pre-trained language model with prompts for temporal knowledge graph completion. *Preprint*, arXiv:2305.07912.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge,

Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2023. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. *arXiv preprint arXiv:2306.11489*.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2024. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *Preprint*, arXiv:2306.11489.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *Preprint*, arXiv:2402.14568.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Towards better dynamic graph learning: New architecture and unified library. *Preprint*, arXiv:2303.13047.

Xuanqing Yu, Wangtao Sun, Jingwei Li, Kang Liu, Chengbao Liu, and Jie Tan. 2024. Onsep: A novel online neural-symbolic framework for event prediction based on large language model. *Preprint*, arXiv:2408.07840.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2023. Back to the future: Towards explainable temporal reasoning with large language models. *Preprint*, arXiv:2310.01074.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. 2024. Dtgb: A comprehensive benchmark for dynamic text-attributed graphs. *Preprint*, arXiv:2406.12072.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2025. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Lixiao Zheng, Jipeng Xiao, Shuai Ma, Zuxi Chen, and Xiangyu Luo. 2025. Temporal cycle enumeration for detecting financial fraud. *DATA INTELLIGENCE*, 7(3):567–590.

Dataset	# Entity	# Relation	Train	Valid	Test
ICEWS14	12,498	260	323,895	-	341,409
ICEWS05-15	10,094	251	368,868	46,302	46,159
ICEWS1819	31,796	266	770,050	165,011	165,010
GDELT	7,691	240	1,734,399	238,765	305,241

Table 7: Statistics of the datasets.

A Dynamic Parameter Generation

Structural recall capacity (k_{struc}) We adapt the number of structurally retrieved nodes to the query node's degree: higher-degree nodes warrant broader search. We compute a scaling factor from the node's log-degree, $\ell_u = \log(d_u + 1)$, normalized by the global mean $\mu_{\text{logdeg}} \in \mathbb{G}$:

$$k_{\text{struc}} = \left\lfloor k_{\text{struc}}^{(\text{base})} \big(1 + \gamma_{\text{struc}}(\ell_u - \mu_{\text{logdeg}})\big) \right\rfloor. \quad (12)$$

Here, γ_{struc} is a sensitivity factor. The result is clamped within a reasonable range $[k_{\min}, 2 \cdot k_{\text{struc}}^{(\text{base})}]$ to ensure stability.

Temporal recall capacity (k_{tem}) We adapt the temporal window to recent activity: when events are sparse, we narrow the look-back to emphasize the most recent nodes. Let Δt_k be the gap from t to the $k_{\text{tem}}^{(\text{base})}$-th most recent event and let $\mu_{\Delta t} \in \mathbb{G}$ denote the global mean inter-event gap. We set

$$\tau = \tanh\left(\frac{\Delta t_k - \mu_{\Delta t}}{\mu_{\Delta t}}\right), \quad (13)$$

$$k_{\text{tem}} = \left\lfloor k_{\text{tem}}^{(\text{base})} \big(1 - \gamma_{\text{tem}}\tau\big) \right\rfloor, \quad (14)$$

where γ_{tem} is the scaling factor for time recall. The $\tanh$ function ensures a smooth and bounded adjustment.

Semantic recall capacity (k_{sem}) We size the semantic pool by the *concentration* of similarities around u. Let σ_u be the standard deviation of cosine similarities between u and its top-$2k_{\text{sem}}^{(\text{base})}$ nearest semantic neighbors. A large σ_u (steep drop-off) favors a *smaller*, more precise pool; a small σ_u (flat tail) favors a *larger* pool. We adjust k_{sem} relative to the global mean $\mu_\sigma \in \mathbb{G}$:

$$\psi = \tanh\left(\frac{\sigma_u - \mu_\sigma}{\mu_\sigma}\right), \quad (15)$$

$$k_{\text{sem}} = \left\lfloor k_{\text{sem}}^{(\text{base})} \left(1 - \gamma_{\text{sem}}\psi\right) \right\rfloor. \quad (16)$$

Here, $k_{\text{sem}}^{(\text{base})}$ is the default size; σ_u measures local similarity spread; μ_σ provides global calibration; $\gamma_{\text{sem}} \in (0, 1]$ controls adjustment strength.

Scoring weight α α balances the influence of structural vs. semantic evidence in the similarity calculation, which is tuned based on the node's structural connectivity. Well-connected nodes (high ℓ_u) provide more reliable structural signals, so α is increased to more strongly weigh structural similarity:

$$z_\alpha = \tanh\left(\frac{\ell_u - \mu_{\text{logdeg}}}{\phi_\alpha}\right), \quad (17)$$

$$\alpha = \text{clip}\left(\alpha^{(\text{base})} + \delta_\alpha z_\alpha, 0.05, 0.95\right). \quad (18)$$

Here, δ_α controls the maximum adjustment range, and ϕ_α is a smoothing factor.

Scoring weight β. β balances the historical consistency score (S_{hist}) against the self-similarity score (S_{self}), which is adapted according to the maturity of the query node, defined by its number of past interactions (n_u). Nodes with extensive history provide more reliable data for the historical partner score, so β is increased. For nodes with sparse history, the model should rely more on the self-similarity:

$$z_\beta = \tanh\left(\frac{n_u - \mu_{\text{hist}}}{\phi_\beta}\right), \quad (19)$$

$$\beta = \text{clip}\left(\beta^{(\text{base})} + \delta_\beta z_\beta, 0.05, 0.95\right). \quad (20)$$

Here, $\mu_{\text{hist}} \in \mathbb{G}$ is the average number of historical interactions per node, δ_β is the adjustment range, and ϕ_β is a smoothing factor.

B Details of Datasets

GDELT [5] is constructed from the Global Database of Events, Language, and Tone (GDELT) project, which monitors political events and activities across the world in near real-time. Nodes correspond to political actors (e.g., United States, Kim Jong Un) and are represented by their names. Edges denote types of interaction or relationship between these actors (e.g., MAKE_STATEMENT, ENGAGE_IN_DIPLOMACY). The textual attributes of edges are derived from the verbal descriptions of these relation types. Each event is timestamped with 15-minute granularity, resulting in a high-resolution temporal graph that captures rapidly evolving political dynamics.

ICEWS [6] is derived from the Integrated Crisis Early Warning System and is commonly used in

[5] https://www.gdeltproject.org/

[6] https://dataverse.harvard.edu/dataverse/icews

temporal knowledge graph research. We consider three standard variants: **ICEWS14**, **ICEWS05–15**, and **ICEWS18–19**, which differ in their temporal coverage. ICEWS14 contains events from 2014, ICEWS05–15 spans the period from 2005 to 2015, and ICEWS18–19 covers events from January 1, 2018, to December 31, 2019. In all variants, nodes represent political entities and are associated with composite textual attributes such as name, sector, and nationality. Relations correspond to discrete political or military event types, with edge text derived from their semantic descriptions. Events are temporally ordered at a daily granularity. Compared to GDELT, ICEWS datasets exhibit a coarser temporal resolution (24-hour intervals) and a substantially larger node set, resulting in a sparser and more diverse interaction structure.

C Details of Baselines

C.1 Temporal Graph Models

JODIE (Kumar et al., 2019) models dynamic graphs using coupled recurrent networks to update entity embeddings over time. A projection mechanism extrapolates future embedding trajectories to support temporal link prediction.

DyRep (Trivedi et al., 2019) updates node representations after each interaction under a temporal point process formulation. It captures evolving structural patterns via time-aware attention to model graph dynamics.

TGAT (Xu et al., 2020) employs self-attention to aggregate temporal neighborhoods and models time features using functional encodings based on harmonic analysis, enabling inductive learning in continuous-time graphs.

CAWN (Wang et al., 2022) learns inductive node representations by sampling anonymized temporal walks that capture causal interaction patterns, which are then encoded and aggregated by neural networks.

TCL (Wang et al., 2021) adopts a dual-stream Transformer to separately encode temporal neighborhoods of interacting nodes and integrates structural and temporal information through topology-aware attention.

GraphMixer (Cong et al., 2023) demonstrates that simple architectures with fixed time encodings can be effective, combining link encoders, node encoders, and a lightweight prediction head for temporal link prediction.

DyGFormer (Yu et al., 2023) represents nodes via co-occurrence patterns in historical interactions and segments long sequences into patches, enabling efficient modeling of extended temporal context.

DiffuTKG (Cai et al., 2024) reformulates temporal knowledge graph reasoning as a conditional diffusion process. It encodes historical events as conditioning sequences and predicts future facts by iteratively denoising noisy target representations, with an uncertainty regularization mechanism to mitigate bias toward frequent events.

RE-GCN (Li et al., 2021) extends relational graph convolutional networks to temporal settings by recurrently evolving relation-aware parameters over time. It captures temporal dynamics through parameter evolution rather than node state recurrence, enabling modeling of relation-specific temporal patterns.

TiRGN (Li et al., 2022) models temporal knowledge graphs using a recurrent graph neural network with relation-specific temporal gating. It integrates structural aggregation with time-aware recurrence to capture evolving interaction patterns for temporal link prediction.

C.2 LLM-based Methods

All LLM-based baselines use Qwen3-8B as the backbone model. Qwen3 is an open-source large language model released by Alibaba in 2025, featuring a Mixture-of-Experts architecture and a mixture inference mechanism that balances deep reasoning and efficient generation.

ICL (Lee et al., 2023) casts temporal knowledge graph forecasting as an in-context learning task, retrieving relevant historical facts, constructing structured prompts, and ranking candidate nodes based on LLM output probabilities without task-specific training.

GAD (Lei et al., 2025) introduces a multi-agent LLM framework for dynamic text-attributed graphs, combining global and local summary agents with a knowledge reflection mechanism to support adaptive and transferable reasoning.

AnRe (Tang et al., 2025) is a training-free reasoning approach that retrieves similar historical

events via semantic clustering and constructs contexts using both long-term and short-term histories. It leverages LLM-generated analogical examples to enable few-shot prediction of future events.

D Random Projection

The Random Projection Module is responsible for generating dynamic structural node embeddings without training (Lu et al., 2024). It maintains a set of stateful projection buffers $\mathbf{P}^{(l)} \in \mathbb{R}^{N\times d}$ for each layer l (where N is the number of nodes and d is the projection dimension). The module is updated incrementally with each new batch of events (u_i, v_i, t_i).

For a new interaction (u, v, t), the update for the higher-order projections ($l \geq 1$) applies a time-decayed message passing step:

$$\eta = \exp(-\lambda(t - t_{\text{prev}})), \tag{21}$$

$$\mathbf{P}^{(l)}[u] \leftarrow \mathbf{P}^{(l)}[u] + \eta\mathbf{P}^{(l-1)}[v], \tag{22}$$

$$\mathbf{P}^{(l)}[v] \leftarrow \mathbf{P}^{(l)}[v] + \eta\mathbf{P}^{(l-1)}[u], \tag{23}$$

where λ is a time decay factor and t_{prev} is the time of the last update. The final structural embedding for a node is the concatenation of its projections across all layers: $\mathbf{e}_{\text{struc}} = \|_{l=0}^{L}\mathbf{P}^{(l)}[u]$. In the experimental setup, the structural vector dimension factor is set to 10, the maximum hop count for random walks is set to 3, and the time decay weight is $1e-7$.

E LLM-Based Scoring and Decision Functions

Given a prompt P constructed from retrieved evidence and optional few-shot demonstrations, the LLM produces a conditional token distribution over the vocabulary:

$$p_\theta(y \mid P) = \text{softmax}(z_\theta(P)), \tag{24}$$

where $z_\theta(P)$ denotes the pre-softmax logits of the model.

Link Prediction For link prediction, we formulate inference as a binary choice between a positive candidate o^+ and a negative candidate o^-. The LLM is prompted to output a decision token $y \in \{\texttt{A}, \texttt{B}\}$, corresponding to (u, r, o^+, T) and (u, r, o^-, T) respectively. The preference score is computed as:

$$s(o^+, o^-) = p_\theta(y = \texttt{A} \mid P), \tag{25}$$

with symmetric score calibration applied by swapping option order. Aggregating such pairwise scores over negatives induces a global ranking used to compute AUC and AP.

Node Retrieval For node retrieval, each candidate node v_i is evaluated independently against the query context via a pointwise comparison. The final retrieval score for v_i is obtained by aggregating its preference probabilities across comparisons:

$$S(v_i) = \mathbb{E}_{v_j\sim\mathcal{N}}\left[p_\theta(v_i \succ v_j \mid P)\right], \tag{26}$$

where $\mathcal{N}$ denotes the negative candidate set. Candidates are ranked by $S(v_i)$ to compute Hit@K and related metrics.

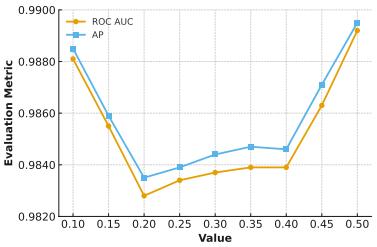
F Additional Implementation Details

Parameter Setting We choose a set of default values as anchors for the dynamic parameter module. Unless specified otherwise, the baseline recall budgets are $k_{str} = k_{tem} = k_{sem} = 55$, the fusion weights are $\alpha = \beta = 0.5$, and the time-decay rate for historical partner weighting is $\lambda = 0.01$. We set the future positive ratio to 0.15 and the negative ratio to 0.5 based on heuristic scores. To balance context length and history coverage, we include 100 golden positives and use 3-shot demonstrations in the few-shot module.

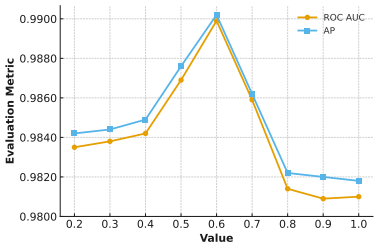
Negative Sampling Strategy For each pointwise decision in the link prediction task, the LLM compares the ground-truth option v_{true} with a sampled negative option v_{neg}. We consider two strategies: (i) **Global sampling** (default), where v_{neg} is drawn uniformly from all nodes excluding u and v_{true}; and (ii) **Recall-pool sampling**, where v_{neg} is drawn from the recall candidate set $\mathcal{C}$, yielding harder and more informative negatives.

Position Bias Calibration We follow the pairwise setup of DTGB for link prediction. For LLM methods, We compute confidence from the token probabilities of the choice markers ("A" and "B") at the final position, rather than relying on free-form outputs. To reduce positional bias, we run two prompts with swapped option orders: (1) A:v_{true}, B:v_{neg} and (2) A:v_{neg}, B:v_{true}. Let $P_1(A)$ denote the probability of choosing A in pass 1 and $P_2(B)$ the probability of choosing B in pass 2. The calibrated confidence for v_{true} is

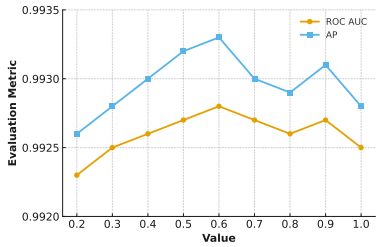
$$P_{\text{final}}(v_{\text{true}}) = \frac{P_1(A) + P_2(B)}{2}. \tag{27}$$



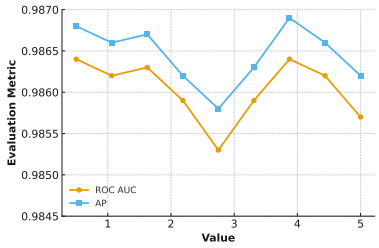
(a) Performance of different sensitivity factors γ_{struc}.



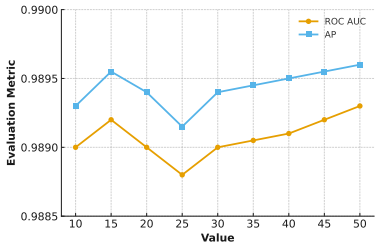
(b) Performance of different sensitivity factors γ_{sem}.



(c) Performance of different sensitivity factors γ_{time}.



(d) Performance of different smoothing factors ϕ_α.



(e) Performance of different smoothing factors ϕ_β.

Figure 6: Hyperparameter sensitivity analysis under ICEWS1819.

Online Protocol All evaluations are conducted in a streaming setting, where test queries are processed sequentially and ground-truth labels become available after each prediction, consistent with the online temporal reasoning protocol in DTGB. All baseline results are either directly taken from DTGB or reproduced under the same task definitions, negative sampling strategies, and evaluation metrics to ensure strict protocol alignment.

G Hyperparameter Sensitivity Analysis

Figure 6 reports AP/AUC under different sensitivity and smoothing factors, with other parameters fixed to their default values.

Impact of γ_{struc}. γ_{struc} controls the degree-aware scaling of structural recall. Small values lead to nearly fixed budgets and under-utilize structural evidence for high-degree nodes, while moderate values temporarily reduce recall for low-degree queries, causing a slight drop in AP/AUC. Larger γ_{struc} allocates sufficient budget to high-degree nodes, where structural signals are informative, yielding performance recovery and improvement.

Impact of γ_{sem}. γ_{sem} adjusts the semantic recall budget. Overly small values limit semantic coverage for low-degree queries, whereas overly large values introduce semantically similar but structurally irrelevant distractors. A moderate setting balances coverage and noise, achieving the best AP/AUC.

Impact of γ_{time}. γ_{time} governs the responsiveness of temporal recall to inter-event sparsity. Small values underreact to recent activity bursts, while moderate values adaptively expand recall during bursts and contract it otherwise, maximizing coverage of true targets. Excessively large values make recall overly sensitive to short-term fluctuations, leading to unstable candidate sets and mild performance degradation.

Impact of ϕ_α. ϕ_α controls the smoothing of the structure-semantic mixing weight α. Low values behave similarly to a global constant and slight increases initially perturb a well-tuned baseline. Moderate smoothing enables α to adapt across degree regimes, improving accuracy, whereas overly large values over-amplify regime differences and reduce robustness.

Impact of ϕ_β. ϕ_β regulates the smoothing of the history-self fusion weight β. Small values make β sensitive to noisy maturity estimates, resulting in oscillatory performance. Larger values stabilize the adaptation, allowing S_{hist} to dominate for mature nodes while preserving S_{self} for sparse ones, leading to consistent performance gains.

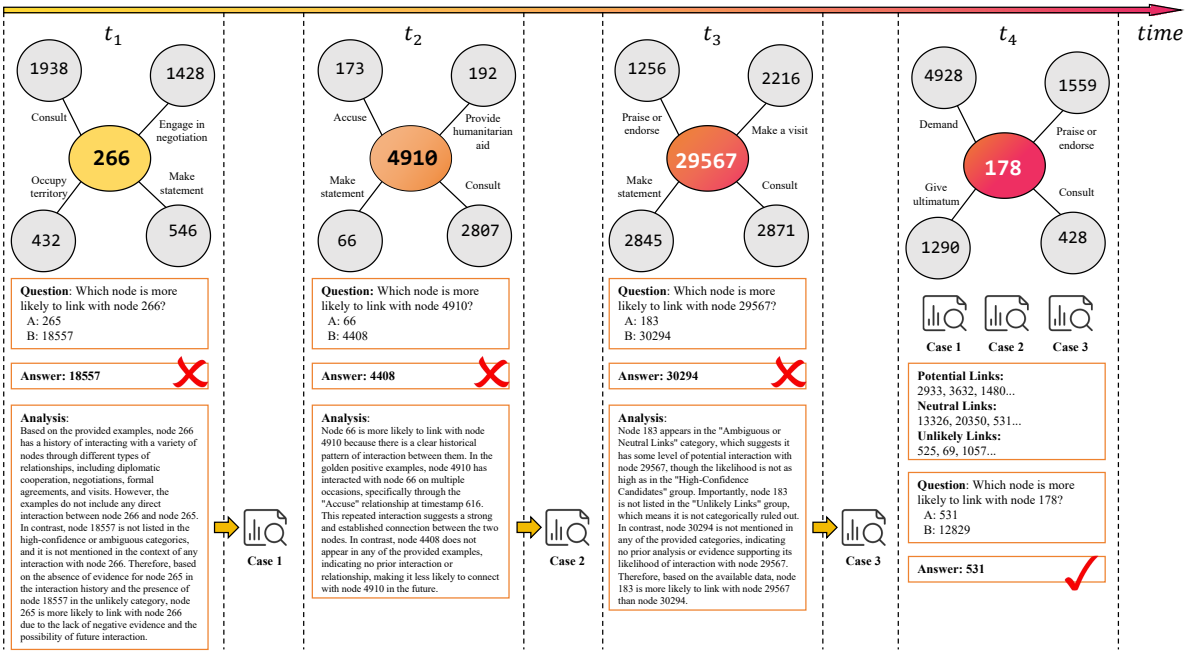


Figure 7: Case study of the self-diagnosis mechanism. Entity names are replaced with unique identifiers to prevent semantic leakage. The examples illustrate error detection, reflective analysis, and case-based reuse at test time.

H Case Study of Self-Diagnosis Reasoning

Figure 7 illustrates how the self-diagnosis module operates on test-time errors using ID-only entities, without any textual semantics. When the model makes an incorrect or low-confidence decision, it is prompted to generate a brief diagnostic rationale conditioned on the same evidence; the resulting explanation is stored as a few-shot case for subsequent inference.

Case 1: Structure-driven correction. For query node u=4910 (choices 66 vs. 4408), the model initially selects the incorrect option. The diagnostic step identifies repeated historical interactions between 4910 and 66 (via the *Accuse* relation at timestamp 616) and the absence of supporting evidence for 4408. This explanation is added to the case library and subsequently biases similar queries toward the history-consistent choice.

Case 2: Evidence tiers as decision cues. For query node u=29567 (choices 183 vs. 30294), the reflective analysis explicitly references the score-based partitions in the prompt. Node 183 appears in the *Ambiguous/Neutral* tier, while 30294 is absent from all tiers, indicating a lack of evidence. The model therefore favors 183, demonstrating how numerical scores are converted into reusable, interpretable cues.

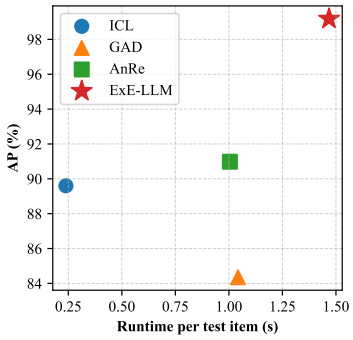


Figure 8: Comparison between model performance and per-item runtime.

Discussion. These cases show that ExE-LLM can refine its decisions from structural history and score-tier evidence, even without entity semantics. By caching error-driven rationales as few-shot examples, the framework forms a closed-loop improvement mechanism at test time. We expect that incorporating lightweight textual descriptors would further reduce ambiguity, while the self-diagnosis mechanism itself remains unchanged.

I Complexity and Runtime Analysis

Complexity of ExE-LLM. For a query node with degree d_u, let P denote the number of ef-

fective historical partners, K the size of the candidate pool after multi-path recall, and $d_{\text{sem}}, d_{\text{str}}$ the semantic and structural embedding dimensions. Adaptive hyperparameter computation relies on constant-time statistics and costs $O(1)$ per query. Multi-path recall consists of bounded temporal scanning, structural lookup, and semantic ANN search, yielding $O(k_{\text{time}} + k_{\text{struc}} + k_{\text{sem}}) = O(K)$. Candidate scoring aggregates similarities against P historical partners and the query itself, leading to a dominant cost of

$$O\big(KP(d_{\text{sem}} + d_{\text{str}}) + K \log K\big),$$

where the second term accounts for candidate partitioning. Streaming updates of the random projection module incur $O(L\, d_{\text{str}})$ per event, and adjacency/history maintenance is constant time. The self-diagnosis step is triggered sparsely and has bounded amortized overhead due to the capped few-shot buffer. Since K and P are adaptively bounded, the effective computation scales sublinearly with the graph size.

Runtime–Accuracy Trade-off. Figure 8 compares per-item runtime and accuracy across test-time methods. ExE-LLM achieves the highest AP (99.18) at 1.48 s per item, outperforming ICL and AnRe by large margins. Although slower than prompt-only baselines at inference, ExE-LLM is entirely training-free and avoids the substantial offline cost of multi-epoch training required by DGNNs. As a result, ExE-LLM occupies a favorable point on the runtime-accuracy frontier, particularly for dynamic settings where rapid test-time adaptation is preferred over amortized training efficiency.

Algorithm 1 Heuristic Evidence Synthesizer (HES)

Require: Query $q = (u, r, ?, T)$; history $\mathcal{H}$ (events with timestamps); adjacency Adj; node history Hist; semantic embeddings $\mathbf{Z}^{sem}$; structural embeddings $\mathbf{Z}^{str}$; (optional) relation embeddings $\mathbf{Z}^{rel}$; dynamic hyperparameters $\Theta = \{k_{str}, k_{tem}, k_{sem}, \alpha, \beta\}$; time-decay λ; partition ratios (p, q); option set $\mathcal{O}$ (true option + negatives)

Ensure: Multi-evidence text $\mathcal{E}$; pointwise prompts $\{\mathcal{P}(o)\}_{o \in \mathcal{O}}$

Phase I: Multi-path Recall

$\mathcal{V}_{str} \leftarrow \mathrm{TopK\_CommonNbr}(u, k_{str}, \mathrm{Adj})$

$\mathcal{V}_{tem} \leftarrow \mathrm{TopK\_RecentActive}(T, k_{tem}, \mathcal{H})$

$\mathcal{V}_{sem} \leftarrow \mathrm{TopK\_SemanticNN}(u, k_{sem}, \mathbf{Z}^{sem})$

$\mathcal{C} \leftarrow \mathcal{V}_{str} \cup \mathcal{V}_{tem} \cup \mathcal{V}_{sem}$

Phase II: Heuristic Ranking

$S(\cdot) \leftarrow \mathrm{ScoreCandidates}(u, T, \mathcal{C}, \mathrm{Hist}, \mathbf{Z}^{sem}, \mathbf{Z}^{str}, \mathbf{Z}^{rel}, \lambda, \alpha, \beta)$ ▷ `ScoreCandidates` implements the bidirectional heuristic in §3.2

$\pi \leftarrow \mathrm{SortDesc}(\mathcal{C}, S)$ ▷ ranking order

Phase III: Category Partition (for evidence synthesis)

$(\mathcal{C}^+, \mathcal{C}^\circ, \mathcal{C}^-) \leftarrow \mathrm{Partition}(\pi, p, q)$ ▷ $\mathcal{C}^+$: potential future, $\mathcal{C}^\circ$: ambiguous, $\mathcal{C}^-$: unlikely

Phase IV: Graph-to-Text Evidence Construction

$\mathcal{E}_A \leftarrow \mathrm{SerializeGoldenHistory}(u, T, \mathrm{Hist})$

$\mathcal{E}_B \leftarrow \mathrm{SerializeCategory}(\mathcal{C}^+, \text{“Potential Future Links”})$

$\mathcal{E}_C \leftarrow \mathrm{SerializeCategory}(\mathcal{C}^\circ, \text{“Ambiguous Links”})$

$\mathcal{E}_D \leftarrow \mathrm{SerializeCategory}(\mathcal{C}^-, \text{“Unlikely Links”})$

$\mathcal{E} \leftarrow \mathrm{Concat}(\mathcal{E}_A, \mathcal{E}_B, \mathcal{E}_C, \mathcal{E}_D)$

Phase V: Pointwise Additional Context for each option

for $o \in \mathcal{O}$ **do**

  $c_{uo} \leftarrow |\mathcal{N}_{<T}(u) \cap \mathcal{N}_{<T}(o)|$ ▷ common neighbors before T

  $f_{uo} \leftarrow \sum_{(x,*,t) \in \mathrm{Hist}(o)} \mathbb{I}[x = u \wedge t < T]$ ▷ past interaction count

  $\rho^{sem}_{uo} \leftarrow \cos(\mathbf{Z}^{sem}_u, \mathbf{Z}^{sem}_o)$

  $\mathcal{E}_{add}(o) \leftarrow \mathrm{Format}(\, c_{uo}, f_{uo}, \rho^{sem}_{uo}\,)$

  $\mathcal{P}(o) \leftarrow \mathrm{BuildPointwisePrompt}\big(\mathcal{E}, \mathcal{E}_{add}(o), (u, r, o, T)\big)$

end for

return $\mathcal{E}$ and $\{\mathcal{P}(o)\}_{o \in \mathcal{O}}$

Setting	Model	ICEWS14		ICEWS05-15		ICEWS1819		GDELT	
		AP	AUC	AP	AUC	AP	AUC	AP	AUC
tr.	**JODIE**	95.32 ± 0.29	94.94 ± 0.29	97.78 ± 0.22	97.61 ± 0.19	98.24 ± 0.62	98.21 ± 0.95	94.82 ± 0.18	95.62 ± 0.27
	DyRep	94.88 ± 0.45	94.33 ± 0.50	96.80 ± 0.20	96.56 ± 0.23	98.13 ± 0.29	97.99 ± 0.39	94.15 ± 0.13	94.77 ± 0.11
	TGAT	97.65 ± 0.08	97.42 ± 0.10	98.94 ± 0.03	98.85 ± 0.02	98.05 ± 0.52	97.87 ± 0.65	93.42 ± 0.39	93.41 ± 0.46
	CAWN	96.79 ± 0.14	96.12 ± 0.17	98.45 ± 0.08	98.16 ± 0.10	98.38 ± 0.31	98.15 ± 0.41	93.98 ± 0.36	94.19 ± 0.26
	TCL	97.70 ± 0.07	97.45 ± 0.06	99.06 ± 0.02	98.93 ± 0.02	98.66 ± 0.17	98.42 ± 0.36	95.54 ± 0.06	95.71 ± 0.07
	GraphMixer	96.31 ± 0.07	95.97 ± 0.08	98.34 ± 0.03	98.19 ± 0.03	96.10 ± 0.85	93.99 ± 0.79	92.99 ± 0.28	93.16 ± 0.21
	DyGFormer	97.77 ± 0.06	97.34 ± 0.07	99.08 ± 0.02	98.93 ± 0.03	98.84 ± 0.13	98.65 ± 0.24	96.40 ± 0.02	96.48 ± 0.07
	RE-GCN	96.26 ± 0.14	95.99 ± 0.15	97.91 ± 0.06	97.77 ± 0.06	98.39 ± 0.25	98.35 ± 0.24	95.66 ± 0.08	95.72 ± 0.07
	TiRGN	96.20 ± 0.08	95.99 ± 0.07	97.45 ± 0.14	97.27 ± 0.15	97.85 ± 0.20	97.84 ± 0.18	94.80 ± 0.52	94.87 ± 0.59
	DiffuTKG	97.11 ± 0.02	96.90 ± 0.02	98.70 ± 0.03	98.57 ± 0.03	98.77 ± 0.07	98.71 ± 0.07	94.80 ± 0.10	95.52 ± 0.03
	ICL	86.11 ± 0.05	85.47 ± 0.02	90.10 ± 0.01	90.08 ± 0.03	89.60 ± 0.04	88.63 ± 0.04	90.56 ± 0.06	90.33 ± 0.07
	GAD	85.60 ± 1.01	83.34 ± 1.05	86.77 ± 0.87	84.96 ± 0.74	84.35 ± 1.03	82.48 ± 1.03	85.56 ± 0.64	84.71 ± 0.74
	AnRe	86.45 ± 0.04	86.34 ± 0.02	91.88 ± 0.07	91.76 ± 0.08	90.98 ± 0.09	90.00 ± 0.08	83.71 ± 0.02	81.86 ± 0.01
	ExE-LLM	**98.01 ± 0.04**	**97.94 ± 0.03**	**99.15 ± 0.05**	**99.02 ± 0.06**	**99.18 ± 0.03**	**99.15 ± 0.03**	**97.69 ± 0.08**	**97.73 ± 0.09**
in.	**JODIE**	85.48 ± 0.26	84.41 ± 0.36	93.73 ± 0.35	92.83 ± 0.32	93.82 ± 0.71	91.15 ± 0.81	90.72 ± 0.17	89.77 ± 0.35
	DyRep	84.61 ± 0.56	83.88 ± 0.54	91.24 ± 0.43	90.92 ± 0.43	94.44 ± 0.37	93.90 ± 0.54	87.56 ± 0.22	87.91 ± 0.02
	TGAT	92.92 ± 0.12	92.64 ± 0.11	96.58 ± 0.08	96.24 ± 0.09	91.81 ± 0.56	91.51 ± 0.61	75.00 ± 0.67	75.01 ± 0.74
	CAWN	90.50 ± 0.41	88.50 ± 0.43	95.51 ± 0.24	94.44 ± 0.28	94.06 ± 0.28	93.30 ± 0.76	79.80 ± 0.16	79.09 ± 0.10
	TCL	92.44 ± 0.15	91.85 ± 0.18	97.24 ± 0.06	96.82 ± 0.09	95.36 ± 0.18	94.71 ± 0.11	84.30 ± 0.53	85.44 ± 0.45
	GraphMixer	88.02 ± 0.20	87.64 ± 0.15	94.81 ± 0.10	94.24 ± 0.11	88.11 ± 0.76	88.58 ± 0.89	72.98 ± 0.60	73.61 ± 0.58
	DyGFormer	92.84 ± 0.13	91.35 ± 0.12	97.39 ± 0.05	96.85 ± 0.04	96.65 ± 0.11	96.13 ± 0.10	91.72 ± 0.14	91.35 ± 0.24
	RE-GCN	88.60 ± 0.34	88.20 ± 0.41	94.00 ± 0.13	93.80 ± 0.13	94.84 ± 0.75	94.78 ± 0.73	82.69 ± 0.43	82.19 ± 0.36
	TiRGN	89.09 ± 0.30	88.95 ± 0.36	92.58 ± 0.43	92.32 ± 0.43	93.95 ± 0.54	94.10 ± 0.45	80.92 ± 1.46	80.22 ± 1.53
	DiffuTKG	90.07 ± 0.32	89.42 ± 0.34	96.08 ± 0.03	95.66 ± 0.05	96.04 ± 0.17	95.80 ± 0.19	83.60 ± 0.08	84.96 ± 0.06
	ICL	86.04 ± 0.01	85.44 ± 0.03	89.97 ± 0.02	89.99 ± 0.05	88.33 ± 0.06	87.26 ± 0.05	90.02 ± 0.01	89.76 ± 0.01
	GAD	84.77 ± 1.44	82.78 ± 1.25	85.23 ± 0.70	83.32 ± 0.65	83.50 ± 0.02	81.59 ± 0.02	84.94 ± 2.25	84.12 ± 2.40
	AnRe	85.12 ± 0.02	84.86 ± 0.03	90.97 ± 0.03	90.75 ± 0.06	90.47 ± 0.16	89.36 ± 0.19	80.78 ± 0.12	78.43 ± 0.05
	ExE-LLM	**97.99 ± 0.03**	**97.91 ± 0.05**	**99.06 ± 0.02**	**98.96 ± 0.03**	**99.14 ± 0.06**	**99.13 ± 0.05**	**98.85 ± 0.04**	**98.83 ± 0.02**

Table 8: Under the global negative sampling strategy, the performance comparison of various models on the link prediction task. *tr.* means transductive setting, and *in.* means inductive setting. The results of the dynamic graph models, TKG-specific models, and training-free LLM methods are respectively presented in the white, gray, and blue tables. The best performance within each setting is highlighted in **bold**.

Setting	Model	ICEWS14			ICEWS05-15			ICEWS1819			GDELT		
		Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
tr.	**JODIE**	50.71 ± 1.89	72.06 ± 1.13	85.94 ± 0.70	66.19 ± 1.50	84.36 ± 0.81	93.60 ± 0.45	66.03 ± 0.10	85.32 ± 0.12	-	28.90 ± 0.31	57.52 ± 0.35	86.75±10.01
	DyRep	45.83 ± 2.37	67.37 ± 1.59	83.19 ± 1.46	34.65 ± 6.18	65.36 ± 5.63	87.05 ± 1.42	61.33 ± 0.26	84.15 ± 0.61	-	31.12 ± 0.29	57.31 ± 0.28	83.99 ± 0.37
	TGAT	63.29 ± 0.28	81.00 ± 0.35	91.98 ± 0.32	77.24 ± 0.32	90.55 ± 0.64	96.49 ± 0.41	78.09 ± 0.31	91.88 ± 0.34	-	41.35 ± 0.13	66.21 ± 0.19	88.17 ± 0.35
	CAWN	47.86 ± 2.63	53.40 ± 3.39	56.53 ± 3.57	58.21 ± 4.94	62.38 ± 5.53	64.88 ± 5.73	78.12 ± 0.51	89.51 ± 0.78	-	42.53 ± 0.10	66.31 ± 0.17	87.47 ± 0.74
	TCL	65.28 ± 0.32	81.70 ± 0.25	**92.07 ± 0.41**	**80.91 ± 0.17**	**91.85 ± 0.12**	**96.83 ± 0.04**	**81.88 ± 1.21**	**93.56 ± 1.06**	-	43.53 ± 0.24	67.56 ± 0.29	88.75 ± 0.30
	GraphMixer	53.94 ± 0.20	74.42 ± 0.16	88.37 ± 0.18	70.07 ± 0.41	86.42 ± 0.21	94.71 ± 0.10	80.03 ± 0.78	92.31 ± 0.13	-	38.75 ± 0.33	63.51 ± 0.30	86.78 ± 0.76
	DyGFormer	69.61 ± 0.11	81.26 ± 0.18	86.12 ± 0.22	78.01 ± 0.32	86.90 ± 0.24	91.30 ± 0.23	80.36 ± 0.02	91.75 ± 0.06	-	**47.64 ± 0.08**	**70.89 ± 0.10**	**90.16 ± 0.11**
	RE-GCN	40.61 ± 0.26	73.33 ± 0.21	86.66 ± 1.00	66.96 ± 0.29	81.26 ± 0.90	89.31 ± 1.45	OOM	OOM	OOM	OOM	OOM	OOM
	TiRGN	45.60 ± 0.83	77.29 ± 1.74	85.23 ± 1.95	64.97 ± 1.03	84.97 ± 1.03	89.95 ± 0.54	OOM	OOM	OOM	OOM	OOM	OOM
	DiffuTKG	53.07 ± 2.03	78.29 ± 2.04	87.48 ± 2.06	70.39 ± 0.95	86.63 ± 0.95	91.85 ± 0.94	OOM	OOM	OOM	OOM	OOM	OOM
	ICL	30.77 ± 0.07	35.77 ± 0.04	43.22 ± 0.14	36.05 ± 0.05	47.09 ± 0.09	50.02 ± 0.13	35.10 ± 0.09	39.20 ± 0.10	46.31 ± 0.09	16.91 ± 0.03	21.01 ± 0.37	29.09 ± 0.45
	GAD	40.45 ± 0.41	45.10 ± 0.32	49.88 ± 0.52	43.10 ± 0.98	49.37 ± 0.45	54.87 ± 0.08	44.62 ± 0.38	46.45 ± 0.47	49.19 ± 0.17	27.95 ± 0.48	31.50 ± 0.43	36.71 ± 0.64
	AnRe	59.26 ± 0.09	63.11 ± 0.06	67.09 ± 0.09	64.77 ± 0.75	71.18 ± 0.04	75.12 ± 0.32	62.17 ± 0.11	64.61 ± 0.08	66.50 ± 0.16	38.18 ± 0.17	50.11 ± 0.18	52.51 ± 0.91
	ExE-LLM	**71.80 ± 0.02**	**82.63 ± 0.05**	88.50 ± 0.09	77.93 ± 0.04	86.93 ± 0.06	91.83 ± 0.05	75.93 ± 0.02	87.13 ± 0.04	**93.00 ± 0.05**	44.10 ± 0.14	65.77 ± 0.21	84.97 ± 0.11
in.	**JODIE**	24.56 ± 0.91	40.27 ± 0.99	60.64 ± 0.64	45.42 ± 1.21	66.79 ± 0.80	82.37 ± 0.54	51.34 ± 0.10	71.40 ± 0.86	-	30.80 ± 0.74	52.94 ± 0.28	73.44 ± 0.64
	DyRep	20.04 ± 1.05	35.98 ± 0.57	57.47 ± 1.08	22.59 ± 3.64	48.75 ± 3.84	74.97 ± 1.64	47.99 ± 0.26	70.10 ± 0.32	-	27.40 ± 0.51	51.79 ± 0.68	72.66 ± 0.61
	TGAT	37.35 ± 0.33	57.10 ± 0.42	77.47 ± 0.38	56.64 ± 0.41	76.64 ± 0.63	89.83 ± 0.54	57.52 ± 0.31	78.35 ± 0.29	-	26.30 ± 0.42	48.18 ± 0.55	73.29 ± 0.29
	CAWN	36.30 ± 1.94	44.49 ± 2.86	49.48 ± 3.27	49.29 ± 3.97	55.59 ± 5.11	59.25 ± 5.36	63.42 ± 0.51	81.20 ± 0.66	-	29.16 ± 0.11	46.67 ± 0.61	69.25 ± 0.19
	TCL	40.77 ± 0.55	57.99 ± 0.45	75.63 ± 0.41	64.50 ± 0.16	80.53 ± 0.24	90.04 ± 0.15	60.59 ± 1.22	80.26 ± 0.91	-	31.99 ± 0.31	50.90 ± 0.24	75.27 ± 0.11
	GraphMixer	25.85 ± 0.84	44.13 ± 0.61	66.47 ± 0.32	47.25 ± 0.58	68.12 ± 0.52	84.17 ± 0.35	61.14 ± 0.78	80.01 ± 0.88	-	26.10 ± 0.46	47.55 ± 0.81	71.23 ± 0.46
	DyGFormer	50.18 ± 0.34	65.11 ± 0.65	74.60 ± 0.32	63.94 ± 0.45	78.32 ± 0.32	86.12 ± 0.24	63.40 ± 0.28	80.17 ± 0.71	-	36.81 ± 0.38	56.66 ± 0.26	78.44 ± 0.18
	RE-GCN	34.64 ± 0.21	54.26 ± 0.11	65.41 ± 1.25	46.78 ± 1.22	66.32 ± 1.08	76.13 ± 2.20	OOM	OOM	OOM	OOM	OOM	OOM
	TiRGN	30.25 ± 0.92	58.25 ± 1.44	72.21 ± 1.98	49.83 ± 0.58	70.84 ± 0.28	83.64 ± 1.38	OOM	OOM	OOM	OOM	OOM	OOM
	DiffuTKG	48.87 ± 2.97	69.20 ± 2.95	79.44 ± 2.94	57.73 ± 1.41	75.04 ± 1.40	85.34 ± 1.41	OOM	OOM	OOM	OOM	OOM	OOM
	ICL	28.24 ± 0.06	33.67 ± 0.03	42.07 ± 0.08	35.14 ± 0.34	43.55 ± 0.08	49.97 ± 0.03	33.51 ± 0.11	37.41 ± 0.13	43.25 ± 0.10	15.48 ± 0.34	19.77 ± 0.41	26.51 ± 0.45
	GAD	37.02 ± 0.25	44.12 ± 0.90	48.42 ± 0.29	43.01 ± 0.76	46.62 ± 0.73	54.00 ± 0.58	42.23 ± 0.17	43.92 ± 0.21	46.94 ± 0.36	19.65 ± 0.56	23.41 ± 0.59	29.45 ± 0.68
	AnRe	54.84 ± 0.11	61.08 ± 0.43	66.63 ± 0.67	62.36 ± 0.12	70.94 ± 0.08	73.43 ± 0.09	59.52 ± 0.13	63.46 ± 0.09	65.27 ± 0.12	35.58 ± 0.78	45.50 ± 0.85	48.60 ± 0.88
	ExE-LLM	**68.10 ± 0.04**	**78.77 ± 0.07**	**86.17 ± 0.02**	**74.87 ± 0.02**	**84.83 ± 0.06**	**90.27 ± 0.10**	**74.00 ± 0.06**	**85.07 ± 0.11**	**91.73 ± 0.04**	**42.93 ± 0.13**	**63.70 ± 0.17**	**82.63 ± 0.22**

Table 9: Under the global negative sampling strategy, the performance comparison of various models on the node retrieval task. *tr.* means transductive setting, and *in.* means inductive setting. OOM means out-of-memory. The results of the dynamic graph models, TKG-specific models, and training-free LLM methods are respectively presented in the white, gray, and blue tables. The best performance within each setting is highlighted in **bold**.

Symbol	Description
$\mathcal{G}$	Temporal knowledge graph (TKG)
$\mathcal{G}_{<t}$	Historical graph before time t
$e_k = (u, r, v, t)$	Temporal fact (quadruple)
u, v	Head (source) and tail (target) entities
r	Relation type
t, T	Timestamp, query time
$\mathcal{E}, \mathcal{R}$	Entity set and relation set
d_u	Degree of node u
$\ell_u = \log(d_u + 1)$	Log-degree (structural popularity) of u
ΔT_u	Temporal recency of node u
σ_u	Semantic ambiguity of node u
$k_{\text{str}}, k_{\text{tem}}, k_{\text{sem}}$	Recall budgets for structural, temporal, and semantic paths
C	Final candidate set after multi-path recall
$K = \lvert C \rvert$	Size of candidate set
$P = \lvert P(u) \rvert$	Number of historical partners of node u
$\mathbf{z}_v^{\text{str}}$	Structural embedding of node v
$\mathbf{z}_v^{\text{sem}}$	Semantic embedding of node v
$\mathbf{z}_r^{\text{rel}}$	Relation embedding of relation r
$d_{\text{str}}, d_{\text{sem}}$	Dimensions of structural and semantic embeddings
w_i	Time-decayed weight of historical interaction i
λ	Temporal decay factor
$s_{\text{src}}(v)$	Source-centric interaction score
$s_{\text{dst}}(v)$	Destination-centric interaction score
$s_{\text{self}}(v)$	Direct self-similarity score
$s_{\text{int}}(v)$	Interaction consistency score
$S(v)$	Final heuristic score of candidate v
α	Weight balancing structural vs. semantic similarity
β	Weight balancing history vs. self-similarity
Θ	Set of adaptive parameters
$P_1(A), P_2(B)$	LLM choice probabilities in two calibrated passes
$P_{\text{final}}(v)$	Calibrated confidence score for candidate v

Table 10: The formal definition of notations used throughout the paper.

Section	Template
Preamble	You are a link prediction expert in a dynamic graph. Based on the examples of past interactions, determine which of the two new nodes is more likely to connect with the query node. In the following quadruple ‘(u, r, v, t)‘ examples, ‘u‘ is the source node ID, ‘r‘ is the text describing the link type, ‘v‘ is the destination node ID, and ‘t‘ is the timestamp of the interaction.
Expert Reasoning Examples	Here are some previous cases with expert analysis. Learn from them: --- Example Start --- Prompt Context: {failed_prompt_template} Expert Analysis: {reasoning_text} --- Example End --- ...
Golden Positive Examples	### Golden Positive Examples (Confirmed Past Interactions) These links have definitely happened: ({u_id}, {r_his}, {v_his}, {t_his}) ... **(one per line)**
Potential Future Links	### Potential Future Links (High-Confidence Candidates) Based on analysis, these links are highly likely to happen in the future: ({u_id}, might interact with, {v_future}, {t_query}) ...
Ambiguous / Neutral Links	### Ambiguous or Neutral Links (Uncertain Candidates) Based on analysis, the likelihood of these links happening is uncertain: ({u_id}, might interact with, {v_neutral}, {t_query}) ...
Unlikely Links	### Unlikely Links (Low-Confidence Candidates) Based on analysis, these links are very unlikely to happen: ({u_id}, interact with, {v_unlikely}, {t_query}) ...
Additional Context	### Additional Context 1. Common Neighbors Count: The number of common neighbors between {u_id} and Candidate {v_true} is {num_true} ... 2. Historical Interaction Frequency: {u_id} and Candidate {v_true} have interacted {t_true} times ... 3. Semantic Similarity Score: The semantic similarity score between {u_id} and Candidate {v_true} is {sem_true}.
Instruction & Question	Now, based on all the examples above, analyze the examples and answer the following question. The correct answer could be either A or B. You must only output the letter of the correct option A or B. Question: Which node is more likely to link with node {u_id}? A: {v_true} B: {v_neg} Answer:

Figure 9: The prompt template for self-diagnosis reasoning.

Section	Template
Preamble	You are a link prediction expert in a dynamic graph. Based on the examples of past interactions, determine which of the two new nodes is more likely to connect with the query node. In the following quadruple ‘(u, r, v, t)‘ examples, ‘u‘ is the source node ID, ‘r‘ is the text describing the link type, ‘v‘ is the destination node ID, and ‘t‘ is the timestamp of the interaction.
Golden Positive Examples	### Golden Positive Examples (Confirmed Past Interactions) These links have definitely happened: ({u_id}, {r_his}, {v_his}, {t_his}) ... **(one per line)**
Potential Future Links	### Potential Future Links (High-Confidence Candidates) Based on analysis, these links are highly likely to happen in the future: ({u_id}, might interact with, {v_future}, {t_query}) ...
Ambiguous / Neutral Links	### Ambiguous or Neutral Links (Uncertain Candidates) Based on analysis, the likelihood of these links happening is uncertain: ({u_id}, might interact with, {v_neutral}, {t_query}) ...
Unlikely Links	### Unlikely Links (Low-Confidence Candidates) Based on analysis, these links are very unlikely to happen: ({u_id}, interact with, {v_unlikely}, {t_query}) ...
Original Question	Now, based on all the examples above, analyze the examples and answer the following question. The correct answer could be either A or B. You must only output the letter of the correct option A or B. Question: Which node is more likely to link with node {u_id}? A: {v_true} B: {v_neg}
Case Analysis	The correct answer was A: {v_true}. Based on the context provided in the prompt, please provide a brief analysis explaining why node {v_true} was the more likely connection. Reasoning:

Figure 10: The prompt template for few-shot example construction.