A Multi-Expert Structural-Semantic Hybrid Framework for Unveiling Historical Patterns in Temporal Knowledge Graphs

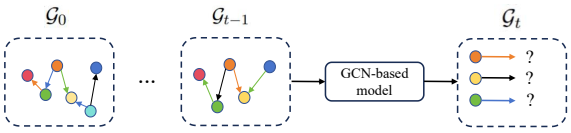
Yimin Deng[1,2], Yuxia Wu[3], Yejing Wang[2], Guoshuai Zhao[1†], Li Zhu[1†], Qidong Liu[1,2]
Derong Xu[2], Zichuan Fu[2], Xian Wu[4†], Yefeng Zheng[5], Xiangyu Zhao[2†], Xueming Qian[1]
[1]Xi'an Jiaotong University, [2]City University of Hong Kong, [3]Singapore Management University
[4]Tencent Jarvis Lab, [5] Westlake University
dymanne@stu.xjtu.edu.cn, guoshuai.zhao@xjtu.edu.cn, zhuli@xjtu.edu.cn
kevinxwu@tencent.com, xianzhao@cityu.edu.hk

Abstract

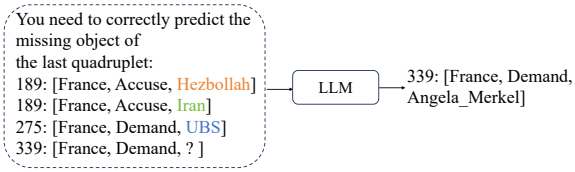
Temporal knowledge graph reasoning aims to predict future events with knowledge of existing facts and plays a key role in various downstream tasks. Previous methods focused on either graph structure learning or semantic reasoning, failing to integrate dual reasoning perspectives to handle different prediction scenarios. Moreover, they lack the capability to capture the inherent differences between historical and non-historical events, which limits their generalization across different temporal contexts. To this end, we propose a **M**ulti-**E**xpert **S**tructural-**S**emantic **H**ybrid (MESH) framework that employs three kinds of expert modules to integrate both structural and semantic information, guiding the reasoning process for different events. Extensive experiments on three datasets demonstrate the effectiveness of our approach.[1]

1 Introduction

Incorporating real-world knowledge is essential for enhancing natural language processing capabilities (Xie et al., 2023; Peng et al., 2023; Pan et al., 2024). Directly extracting specific knowledge or facts from various unstructured texts is time-consuming and laborious, hence knowledge graph (KG) is adopted to store some common facts and reduce retrieval cost. However, traditional KGs are limited to static fact storage. To capture the dynamic nature of facts, temporal knowledge graph (TKG) was proposed to record facts changing over time. It can provide certain evidence for many downstream tasks, like situation analysis, political decision making and service recommendation (Mezni, 2021; Saxena et al., 2021; Jia et al., 2021; Wu et al., 2023, 2024). TKG reasoning aims to predict the missing objects of future events based on existing facts (Leblay and Chekol, 2018; Garcia-Duran et al., 2018; Li et al., 2021), where the formal problem definition is formulated in Section 3.1.



(a) Methods based on structural information.



(b) Methods based on semantic information.

Figure 1: Two research lines of TKG reasoning. One line leverages graph patterns across different timestamps, and the other line utilizes semantic information from event quadruples to capture logical evolution.

According to the type of utilized information, existing methods typically consist of two lines. One line relies on structural information (Figure 1(a)), modeling entity interactions through graph neural network (GCN) (Mo et al., 2021; Cai et al., 2023; Wang et al., 2023a). Previous works in this category have explored the utilization of recurrent networks with neighborhood aggregation mechanisms and recursive graph structures to jointly capture temporal dependencies and concurrent relations in TKGs (Jin et al., 2020; Li et al., 2021). The other line (Figure 1(b)) focus on semantic reasoning with pre-trained language models (PLMs) (Xu et al., 2023a), particularly applying large language models (LLMs) to generate interpretable reasoning (Chang et al., 2024). These methods (Lee et al., 2023; Liao et al., 2024; Luo et al., 2024) typically generate predictions through in-context learning with relevant historical facts. Some methods further enhance performance by incorporating retrieval-

† Corresponding authors.

[1]The code and dataset are available at https://github.com/Applied-Machine-Learning-Lab/MESH-TKGR.

Findings of the Association for Computational Linguistics: ACL 2025, pages 20553–20565

augmented generation and parameter-efficient tuning, allowing LLMs to better adapt to the TKG reasoning task. In summary, existing works focus on either structural or semantic information, overlooking the potential benefits of integrating both types of information. However, different types of information can provide complementary insights for reasoning. The absence of structural information leads to insufficient knowledge of entity interaction patterns, while the lack of semantic information prevents understanding of entities' actions. As exemplified in Figure 1(b), France's recurring "accuse" or "demand" actions could inform predictions about future "demand" action. There is a logical progression from "accuse" to "demand", but the graph-based methods simply treat them as two different relations, losing this reasoning evidence.

From another viewpoint, events to be predicted can be typically divided into two categories: historical events, which have already taken place in the past, and non-historical events, which have never occurred up to now. There is an inherent reasoning gap between different events (Xu et al., 2023b; Gastinger et al., 2024). For historical events, capturing recurrence patterns is crucial while exploring evolution patterns is essential for non-historical events. This inspires several works that attempted to handle historical and non-historical events differently. TiRGN (Li et al., 2022) models both the periodic patterns (which often appear in recurring historical events) and sequential evolution patterns (which characterize non-historical events), while CENET (Xu et al., 2023b) employs a binary classifier to separate the two types of events and focus predictions on relevant candidate entities. These methods often overlook that different types of information have their own advantages when handling different types of events.

To address the aforementioned limitations, we propose a **M**ulti-**E**xpert **S**tructural-**S**emantic **H**ybrid (MESH) framework that effectively integrates structural and semantic information to model historical patterns for temporal knowledge graph reasoning. This model consists of a feature encoder, two kinds of event-aware expert modules and a prediction expert module. The underlying feature encoder outputs structural information from GCN and semantic information from LLM. Then we employ two kinds of event-aware expert modules to learn information weight allocation patterns for historical/non-historical events. There is a challenge in distinguishing event types. To address this, we design a prediction expert module which assigns different weights to each event-aware expert module, thereby implicitly distinguishing different types of events. This unified architecture enables adaptive information fusion without requiring explicit event type labels, offering both flexibility and efficiency. To summarize, the contributions of our work are as follows:

- We discover and verify the complementary advantages of structural and semantic information when applied to different types of events.
- We propose a novel non-generative approach to leveraging LLMs for TKG reasoning, in combination with graph-based models.
- We employ two kinds of event-aware expert modules that adapt to different information preferences between historical and non-historical events, with a prediction expert for automatic weight allocation between experts.
- We conduct extensive experiments on three public benchmarks and the results demonstrate the effectiveness of our proposed method.

2 Related Work

GCN-Based TKG Reasoning Models. GCNs have shown strong ability to model structural information for graphs, leading to a series of GCN-based methods for temporal knowledge reasoning. RE-Net (Jin et al., 2020) utilizes a recurrent event encoder and neighborhood aggregator to capture temporal dependencies and concurrent relations. REGCN (Li et al., 2021) recurrently fit entity and relation features in the order of timestamps. TiRGN (Li et al., 2022) explicitly integrates time embeddings to graph embeddings, which facilitates learning across long temporal periods. GCN-based models treat entities as nodes and integrate representations of entities and relations to predict the next event. While effective in capturing structural patterns, these methods often overlook semantic information in the reasoning process.

LLM-Based TKG Reasoning Models. Some methods formulate TKG reasoning as masked language modeling (MLM) or next-token generation tasks, using language models as the backbone. ChapTER (Peng et al., 2024) uses a pre-trained language model as encoder and employs a prefix-based tuning method to obtain good representations. PPT (Xu et al., 2023a) performs masked token prediction with fine-tuned BERT. Recently, LLMs have demonstrated powerful capabil-

ities in summarization and reasoning. ICL (Lee et al., 2023) leverages LLMs with in-context learning by carefully selecting historical facts as context and decodes the outputs to rank predicted entities. GenTKG (Liao et al., 2024) proposes a retrieval-augmented generation approach with temporal logic rules, while applying parameter-efficient tuning to adapt LLMs for TKG reasoning. CoH (Luo et al., 2024) captures key historical interaction from input text and also applies parameter-efficient tuning. Although these LLM-based methods provide context-based interpretable reasoning, they often struggle to capture the complex structural patterns in TKGs.

3 Methods

In this section, we first introduce the problem definition and notations of the temporal knowledge graph reasoning. Then we present the overall architecture of our proposed approach, followed by a detailed description of each model component.

3.1 Problem Formulation

A TKG $\mathcal{G}$ consists of a set of static graphs $\mathcal{G}_t$, where each static graph contains all the facts at timestamp t. Formally, a TKG can be represented as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the sets of entities, relations, and timestamps, respectively. $\mathcal{F}$ denotes the set of facts, each formulated as a quadruple (s, r, o, t), where $s, o \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{T}$, s/o represents the subject/object and r represents the relation between s and o at t. To model the dynamic nature of real-world knowledge, the TKG reasoning task aims to predict the missing entity in query $(s, r, \_, t)$ with existing facts occur before time t. Additionally, an event (s, r, o, t), if there exists a previous occurrence (s, r, o, k) where $k < t$, is denoted by historical event; otherwise, is denoted by non-historical event.

3.2 Overall Framework

In this section, we briefly introduce the overall architecture of MESH. As shown in Figure 2, it follows a three-layer architecture, consisting of an underlying feature encoder, two kinds of event-aware expert modules in the middle layer, and a prediction expert at the top. Specifically, the feature encoder contains two components: a GCN-based structural encoder taking sub-graph structures as input and a LLM-based semantic encoder taking prompts with entity/relation names as input. These two components capture TKG information from

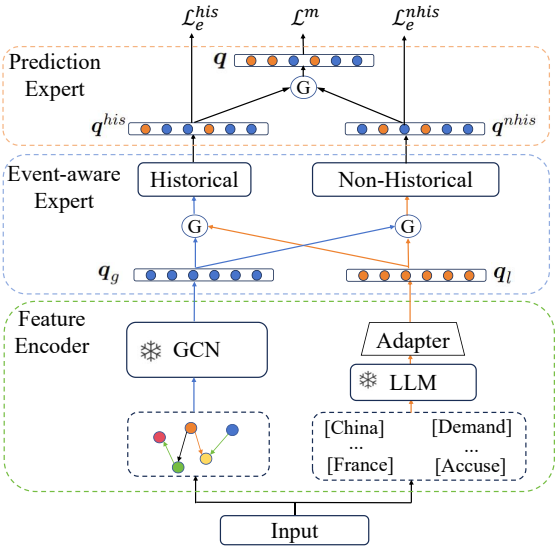


Figure 2: The overall architecture of MESH.

complementary perspectives and generate query representations $\boldsymbol{q}_g$ and $\boldsymbol{q}_s$, respectively. To adaptively handle feature fusion at different layers, we employ query-motivated gates that take $\boldsymbol{q}_g$ as input. The two kinds of specialized event-aware expert modules control the fusion patterns of features for historical or non-historical events, producing representations $\boldsymbol{q}_{his}$/$\boldsymbol{q}_{nhis}$, and the prediction expert learns the representation $\boldsymbol{q}$ for final prediction.

3.3 Feature Encoder

High-quality encoders are essential for integrating and analyzing dynamic knowledge (Zhao et al., 2018a,b; Zhang et al., 2023; Wang et al., 2023b). Since different types of encoders can capture complementary perspectives of TKGs, we employ two independent encoders: a GCN-based encoder for extracting information from graph structures and an LLM-based encoder for modeling semantic information. In this section, we will introduce these two encoders in detail.

3.3.1 Structural Encoder

We employ a structural encoder that learns expressive representations of entity interactions over time. The structural encoder aggregates relational information from the graph topology, enabling the model to incorporate both structural dependencies and temporal dynamics (Xu et al., 2024). Formally, given a temporal knowledge graph $\mathcal{G}$ the structural encoder G generates structural embeddings as:

$$\boldsymbol{H}_g, \boldsymbol{R}_g = G(\mathcal{G}) \quad (1)$$

where $\boldsymbol{H}_g \in \mathbb{R}^{|\mathcal{E}|\times d}$ and $\boldsymbol{R}_g \in \mathbb{R}^{|\mathcal{R}|\times d}$ denote the structural feature of entities and relations, respectively. In this paper, we adopt RE-GCN (Li et al., 2021) as the structural encoder, but our method is not restricted to any specific structural encoder.

3.3.2 Semantic Encoder

Entities and relations in TKGs usually contain rich semantic information. For example, the entity "Javier Solan" is associated with semantic attributes as a Spanish politician, providing valuable contextual knowledge crucial for reasoning tasks. Leveraging this semantic information is essential for enhancing TKG reasoning and improving model performance. Existing methods often leverage the reasoning and generative capabilities of LLMs to directly generate the answer for TKG reasoning tasks (Lee et al., 2023; Liao et al., 2024). However, they typically suffer from high inference latency. To mitigate this issue, recent works have focused on leveraging the representational capacity of LLMs to reduce inference costs (Wang et al., 2023c,d; Liu et al., 2024a). Motivated by this, we adopt an LLM-based approach to encode entities and relations efficiently.

Specifically, we construct the following prompts:

Entity Encoding Template

In the context of <DATA DOMAIN>, please provide <DATA TYPE> background about <ENTITY>.

Relation Encoding Template

In the context of <DATA DOMAIN>, what are the <DATA TYPE> perspectives through which we can understand the <RELATION>?

We fill the underlined places with dataset-specific characteristics and particular entities or relations, then feed these prompts to LLMs such as LLaMA (Touvron et al., 2023) to obtain semantic embeddings. For example, for ICEWS14 dataset, <DATA DOMAIN> and <DATA TYPE> can be political and historical wordings. <ENTITY> can be 'France' and <RELATION> can be 'Abuse'. Finally, we extract the hidden states from the last transformer layers of the LLM to obtain semantic representations of entities and relations, denoted as $\boldsymbol{H}_{LLM} \in \mathbb{R}^{|\mathcal{E}|\times d_{LLM}}$ and $\boldsymbol{R}_{LLM} \in \mathbb{R}^{|\mathcal{R}|\times d_{LLM}}$. However, the original LLM representations are trained for general language tasks (Touvron et al., 2023) and typically have significantly larger dimensions than structural embeddings($d_{LLM} >> d$). To adapt $\boldsymbol{H}_{LLM}, \boldsymbol{R}_{LLM}$ to our TKG reasoning tasks, we employ adapter modules f_H, f_R that compress these representations to a lower-dimensional space, typically implemented as multi-layer perceptrons (MLPs) (Chen et al., 2024; Liu et al., 2024b):

$$\boldsymbol{H}_l = f_H(\boldsymbol{H}_{LLM}), \boldsymbol{R}_l = f_R(\boldsymbol{R}_{LLM}) \quad (2)$$

where $\boldsymbol{H}_l \in \mathbb{H}^{|\mathcal{E}|\times d}, \boldsymbol{R}_l \in \mathbb{R}^{|\mathcal{E}|\times d}$.

3.3.3 Query Representation

From the underlying feature encoder, we obtain entity embeddings $\boldsymbol{h}_g \in \mathbb{R}^d$, $\boldsymbol{h}_l \in \mathbb{R}^d$ and relation embeddings $\boldsymbol{r}_g \in \mathbb{R}^d$, $\boldsymbol{r}_l \in \mathbb{R}^d$ by taking the corresponding rows from $\boldsymbol{H}_g, \boldsymbol{H}_l, \boldsymbol{R}_g, \boldsymbol{R}_l$. We subsequently detail decoders to generate query representations for the query $(s, r, \_, t)$.

Since the convolutional score function has shown its effectiveness as decoder in previous work (Vashishth et al., 2020; Li et al., 2021), we employ ConvTransE as decoder:

$$\boldsymbol{q}_g = D_g(\boldsymbol{h}_g \oplus \boldsymbol{r}_g) \quad (3)$$

$$\boldsymbol{q}_l = D_l(\boldsymbol{h}_l \oplus \boldsymbol{r}_l) \quad (4)$$

where D_g and D_l denote the decoders for structural and semantic features, respectively, $\boldsymbol{q}_g \in \mathbb{R}^d$ and $\boldsymbol{q}_l \in \mathbb{R}^d$ denote the query representations from structural and semantic perspectives.

3.4 Event-Aware Experts

We suggest that different type of events may require different kind of information for reasoning. For example, historical events often involve complex context which are better captured by LLMs, as demonstrated by our analysis in Section 4.6. Existing methods (Li et al., 2021; Liao et al., 2024) overlook this diversity and thus fail to achieve optimal performance on different types of events consistently. This observation motivates us to design a mechanism that can adaptively handle different types of events. Consequently, we propose event-aware experts in this section to adaptively integrate structural and semantic information from previous steps. As shown in Section 4.4, it effectively enhances the model's reasoning capability.

Specifically, we divide events into two categories: historical and non-historical (Xu et al., 2023b). We set M experts for historical events and N experts for non-historical events (Zhang et al., 2024b). We employ $\boldsymbol{q}_g$ as the input to the query-motivated gate, since it captures the evolving

structural patterns of each sub-graph over time and records dynamic structural information, thereby can better distinguish event types. The operation of $i^{th}(i \leq M+N)$ expert is:

$$\alpha_i = \sigma(\boldsymbol{q}_g \mathbf{W}_i + b_i) \tag{5}$$

$$\boldsymbol{q}_i = \alpha_i \cdot \boldsymbol{q}_g + (1-\alpha_i) \cdot \boldsymbol{q}_s \tag{6}$$

where $\mathbf{W}^i \in \mathbb{R}^{d\times 1}$ and b^i denote the weight matrix and bias term of the gating function to generate the weight α_i that indicating the dependency of this expert on structural information. $\boldsymbol{q}_i \in \mathbb{R}^d$ represents the query representation from the i-th expert module, which combines comprehensive views for further prediction. We regard the first M experts as historical experts and the following N as non-historical experts, which are adept in handling historical and non-historical events, respectively.

3.5 Prediction Expert

While we have studied how to leverage different information based on event types, distinguishing the types of events to be predicted remains challenging since the types of future events are unknown. Previous methods (Xu et al., 2023b) typically employ binary classification to determine event types, making the prediction performance susceptible to classification errors. To address this limitation, we design a prediction expert that adaptively integrates information from different kinds of experts without explicit type classification.

Based on query representations produced by multiple event-aware experts, we finally construct a prediction expert to mix all information and predict future events. Similar to Equation (5), we first adaptively allocate weights to experts with the initial query representation $\boldsymbol{q}_g$:

$$\boldsymbol{\alpha} = \sigma(\boldsymbol{q}_g \boldsymbol{W} + \boldsymbol{b}) \tag{7}$$

$$\boldsymbol{q} = \boldsymbol{\alpha} \cdot [\boldsymbol{q}_1, ..., \boldsymbol{q}_{M+N}]^T = \sum_{1\leq i\leq M+N} \alpha_i \boldsymbol{q}_i \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{d\times(M+N)}$ and $\mathbf{b} \in \mathbb{R}^{(M+N)}$ denote the weight matrix and bias term of the gating function, resulting in $\boldsymbol{\alpha} \in \mathbb{R}^{(M+N)}$. Equation (8) combines expert information with the dynamic weights.

The final prediction $\boldsymbol{p}_{s,r,t}$ for query $(s, r, \_, t)$ is made with the matrix product between $\boldsymbol{q}$ and $\boldsymbol{H}_g$:

$$\boldsymbol{p}_{s,r,t} = \sigma(\boldsymbol{q} \cdot \boldsymbol{H}_g) \tag{9}$$

where $\boldsymbol{p}_{s,r,t} \in \mathbb{R}^{|\mathcal{E}|}$ indicates the probability of the missing object is corresponding candidate within the candidate set $\mathcal{E}$.

3.6 Optimization

In this section, we introduce the loss function for the model optimization. There are two training objectives: 1) Event-aware experts should specialize in corresponding event types. 2) The overall prediction is accurate for TKG reasoning tasks.

We can obtain the partial information of historical and non-historical experts as:

$$\boldsymbol{q}^{his} = \sum_{1\leq i\leq M} \alpha_i \boldsymbol{q}_i \tag{10}$$

$$\boldsymbol{q}^{nhis} = \sum_{M+1\leq i\leq M+N} \alpha_i \boldsymbol{q}_i \tag{11}$$

Then, the event-aware predictions can be obtained similar to Equation (9):

$$\boldsymbol{p}^{his}_{s,r,t} = \sigma(\boldsymbol{q}^{his} \cdot \boldsymbol{H}_g) \tag{12}$$

$$\boldsymbol{p}^{nhis}_{s,r,t} = \sigma(\boldsymbol{q}^{nhis} \cdot \boldsymbol{H}_g) \tag{13}$$

By definition, historical facts refer to events that have occurred before t, while non-historical facts have no prior occurrences at time t. We can calculate the frequency of each fact's occurrence before t and construct the historical indicator:

$$F^{s,r}_t(o) = \sum_{k<t} |\{(s,r,o,k)|(s,r,o,k) \in \mathcal{G}_k\}| \tag{14}$$

$$I^{s,r}_t(o) = \begin{cases} 1 \text{ if } F^{s,r}_t(o) > 0 \\ 0 \text{ if } F^{s,r}_t(o) = 0 \end{cases} \tag{15}$$

where $\mathcal{G}_k$ denotes the subgraph at time k. The set $\{(s,r,o,k)|(s,r,o,k) \in \mathcal{G}_k\}$ indicates all events happened at timestamp k that contains the object o. $F^{s,r}_t(o)$ denote the summation of the count over these set prior to the current timestamp t, indicating the frequency of the event. The historical indicator $I^{s,r}_t(o)$ can finally distinguish event types if the frequency is non-zero.

To enable event-aware expert modules to specialize in different event types, we compute the expert loss based on corresponding predictions:

$$\mathcal{L}^{his}_e = -\sum_{(s,r,o,t)\in\mathcal{G}} \boldsymbol{y}_{s,r,t} \boldsymbol{p}^{his}_{s,r,t} I^{s,r}_t(o) \tag{16}$$

$$\mathcal{L}^{nhis}_e = -\sum_{(s,r,o,t)} \boldsymbol{y}_{s,r,t} \boldsymbol{p}^{nhis}_{s,r,t} (1 - I^{s,r}_t(o)) \tag{17}$$

where $\boldsymbol{y}^{s,r}_t \in \mathbb{R}^{|\mathcal{E}|}$ is the one-hot ground truth vector of for entity prediction.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Train	Valid	Test	$\lvert\mathcal{F}_{his}\rvert$	$Rate_{his}$	Δt
ICEWS14	7,128	230	74,845	8,514	7,371	3,064	41.6%	24h
ICEWS18	23,033	256	373,018	45,995	4,9545	20,865	42.1%	24h
ICEWS05-15	10,778	24	368,868	46,302	46,159	24,915	54.0%	24h

Table 1: Statistics of datasets. $\lvert\mathcal{F}_{his}\rvert$ and $Rate_{his}$ denote the number and percentage of historical events in test set, respectively, and Δt denotes the time granularity of each dataset.

Besides, the prediction with all experts should be accurate, resulting in the major loss:

$$\mathcal{L}^m = -\sum_{(s,r,o,t)\in\mathcal{G}} \boldsymbol{y}_{s,r,t}\boldsymbol{p}_{s,r,t} \tag{18}$$

The total loss function is formulated as a weighted sum of the major loss and the auxiliary expert loss:

$$\mathcal{L} = \mathcal{L}^m + \omega(\mathcal{L}_e^{his} + \mathcal{L}_e^{nhis}) \tag{19}$$

where ω is the balancing weight. We detail the training procedure in **Appendix A**.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We conduct experiments on three public benchmarks ICEWS14 (Garcia-Duran et al., 2018), ICEWS18 (Jin et al., 2020) and ICEWS05-15 (Garcia-Duran et al., 2018). Table 1 shows the statistics of these datasets. All these three datasets show a balanced distribution between historical and non-historical facts in their test sets, with historical event ratios ranging from 41.6% (ICEWS14) to 54.0% (ICEWS05-15), making them suitable for evaluating the prediction of different event types.

4.1.2 Evaluation Metrics

Following previous works (Jin et al., 2020; Li et al., 2021), we employ the Mean Reciprocal Rank (MRR) and Hits@k (H@k) as the evaluation metrics. MRR calculates the average of the reciprocal ranks of the first relevant entity retrieved by the model, while H@k calculates the proportion of the correct entity ranked in the top k. The MRR metric is not available for LLM-based methods as they directly generate entities rather than ranking candidates. Detailed definitions of these metrics are shown in **Appendix B.1**.

4.1.3 Baselines

To conduct a comprehensive comparison, we select nine up-to-date TKG reasoning methods, including five graph-based methods and four LLM-based methods. For graph-based models, we choose RE-Net (Jin et al., 2020), REGCN (Li et al., 2021), CENET (Xu et al., 2023b).For LLM-based models, we choose ICL (Lee et al., 2023), CoT (Luo et al., 2024) and GenTKG (Liao et al., 2024).We also compare with two straightforward baselines introduced in **Appendix B.2**, namely "Naive" and "LLM-MLP".

4.1.4 Implementation Details

For the structural encoder G, we employ an efficient graph-based model RE-GCN (Li et al., 2021). The hidden dimension d is 100. The dropout of each GCN layer is set to 0.2. To maintain the stability of structural features, the structural encoder is trained for 500 epochs and then frozen. For the semantic encoder, we utilize LLaMA-2-7B ($d_{LLM} = 4096$) coupled with a two-layer MLP as the adapter. The decoder D_g and D_l are ConvTransE with 50 channels and kernel size of 3. For event-aware experts, $M = N = 1$. All experiments are conducted on an NVIDIA A100 GPU, with the learning rate set to 0.001. Our results are averaged over three random runs.

4.2 Main Results

The experimental results for entity prediction are presented in Table 2. Based on these results, we observe the following findings:

- MESH achieves state-of-the-art performance on ICEWS14 and ICEWS18, surpassing all graph-based and LLM-based baselines. It maintains competitive performance on ICEWS05-15, only second to TiRGN (Li et al., 2022). Moreover, our proposed model can be integrated with any structural or semantic encoder, improving methods like TiRGN as shown in Table 3.
- MESH significantly outperforms our structural encoder RE-GCN on all results. After incorporating semantic information, our model outperforms RE-GCN with improvements of 2.47%/1.55% in MRR, 3.55%/1.63% in H@3, and 2.81%/1.85% in H@10 on ICEWS14/ICEWS18, respectively. These results demonstrate that the strong understanding capability of LLMs can effectively

Model	ICEWS14			ICEWS18			ICEWS05-15		
	MRR	H@3	H@10	MRR	H@3	H@10	MRR	H@3	H@10
Graph-Based Methods									
Naive	-	38.00	44.73	-	4.04	6.29	-	39.66	49.68
RE-Net (Jin et al., 2020)	40.89	45.31	59.23	29.92	32.44	49.51	43.55	48.86	64.78
RE-GCN (Li et al., 2021)	41.89	46.26	61.4	32.41	36.74	52.27	48.42	53.97	68.47
TiRGN (Li et al., 2022)	44.06	49.02	63.84	33.42	37.87	54.11	**49.61**	**55.49**	**69.91**
CENET (Xu et al., 2023b)	39.92	43.56	58.2	27.98	31.67	47.02	42.0	46.99	62.3
LLM-Based Methods									
LLM-MLP	39.77	43.62	58.69	29.25	32.88	47.65	39.85	43.5	58.84
LLM-ICL (Lee et al., 2023)	-	38.9	45.6	-	29.7	36.6	-	49.1	56.6
GenTKG (Liao et al., 2024)	-	48.3	53.6	-	37.25	49.9	-	52.5	68.7
CoH (Luo et al., 2024)	-	46.64	59.11	-	36.22	52.61	-	53.14	68.87
MESH	**44.36**	**49.81**	**64.21**	**33.96**	**38.37**	**54.12**	48.66	54.26	68.57

Table 2: TKG reasoning performance (with time-aware metrics) on ICEWS14, ICEWS18, ICEWS05-15. The best results are in **bold** and the second best results are underlined. Results are averaged over three random runs ($p < 0.05$ under t-test).

enhance the model's power of prediction.

- Compared to LLM-based methods, our approach demonstrates superior performance across all available metrics. It demonstrates the importance of structural information in TKG reasoning. Notably, LLM-based methods show consistently lower scores on H@10 compared to graph-based approaches, revealing their inherent limitations in maintaining comprehensive historical knowledge. This observation aligns with the catastrophic forgetting during continual learning of LLM, indicating their difficulty in effectively leveraging complete historical patterns during reasoning.
- Our method addresses this limitation of LLM-based model by integrating structural information with global information of graph. Moreover, our model significantly reduces the inference cost. While these LLM-based methods usually require over 12 hours for inference on ICEWS14, our approach completes the same task within minutes.

4.3 Compatibility Study

To validate the compatibility of MESH, we conduct experiments with different structural and semantic encoders, as shown in Table 3. Our default configuration employs RE-GCN as the structural encoder G in Equation (1) and LLaMA2-7B as the semantic encoder in Equation (2). For structural encoders, TiRGN shows superior performance over RE-GCN across all metrics. For semantic encoders, Stella-en-1.5B-v5 (Zhang et al., 2024a) slightly outperforms LLaMA2-7B. When integrating these alternative encoders into our framework, we observe consistent improvements. Specifically, replacing RE-GCN with TiRGN leads to better performance, achieving an MRR of 44.97%, H@3 of 50.78%, and H@10 of 65.54%. Incorporating Stella also brings performance improvements. Since structural encoders (e.g., RE-GCN, TiRGN) generally outperform semantic encoders (e.g., LLaMA2, Stella), the improvements are less significant compared to those obtained from better structural encoders. Overall, our method achieves consistent performance gains by integrating both structural and semantic encoders, compared to using a single encoder alone. These experimental results strongly support our claim that our model is not limited to specific structural or semantic encoder, allowing MESH to effectively integrate various advanced encoder modules.

	MRR	H@3	H@10
RE-GCN	41.89	46.26	61.4
TiRGN	44.06	49.02	63.84
LLama-2-7B	39.77	43.62	58.69
Stella-en-1.5B-v5	40.39	44.60	58.96
MESH	44.36	49.81	64.21
MESH w/TiRGN	44.97	**50.78**	65.54
MESH w/Stella	44.53	49.23	63.53
MESH w/TiRGN, Stella	**45.05**	50.56	**65.77**

Table 3: Compatibility study on ICEWS14.

4.4 Ablation Study

Table 4 shows the ablation studies of our proposed model. First, we remove the semantic or structural

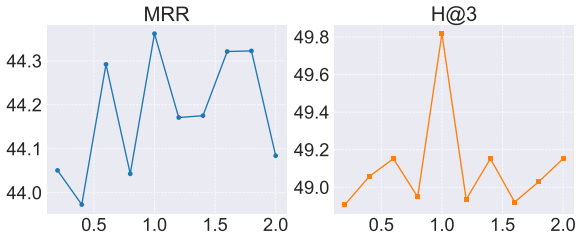


Figure 3: Sensitivity analysis results of ω on ICEWS14.

	MRR	H@3	H@10
w/o Semantic Info	41.89	46.26	61.4
w/o Structural Info	39.77	43.62	58.69
w/o Event-aware	43.96	48.92	64.15
w/o Prediction Expert	43.44	48.28	62.77
MESH	**44.36**	**49.81**	**64.21**

Table 4: Ablation study on ICEWS14.

information obtained from Equation (1)/(2), denoted as **w/o Semantic Info** or **w/o Structural Info**. It leads to a 2.47%/4.59% decrease in MRR, indicating that both types of information are complementary and crucial for accurate predictions. Next, we drop the specialization of event-aware experts as **w/o Event-aware**, specifically by removing the auxiliary expert loss in Equation (19). Finally, we omit the prediction expert, i.e., replace $\boldsymbol{q}$ in Equation (9) with the average of $\{\boldsymbol{q}_i\}$, denoted by **w/o Prediction Expert**. It leads to decreases of 0.92% in MRR, 1.53% in H@3, and 1.54% in H@10, indicating the importance of adaptively integrating multiple event-aware experts. **Appendix C.1** provides the further studies on gating inputs.

4.5 Sensitivity Analysis

To explore the sensitivity of MESH to the expert loss weight ω in Equation (19), we conduct experiments by varying the value of ω from 0.2 to 2.0. As shown in Figure 3, we evaluate the model performance on ICEWS14 using MRR and H@3. The MRR varies between 43.97% and 44.36%, while H@3 varies between 48.91% and 49.81%. The results show that our model maintains stable performance across different values of ω. As ω increases, the model performance first improves and then declines, achieving the best results when $\omega = 1$. This trend indicates that the model performs best when the expert loss and prediction loss are weighted equally, showing that maintaining a balanced scale between these two loss components is better.

Model	Historical			Non-Historical		
	MRR	H@3	H@10	MRR	H@3	H@10
CENET	68.32	72.68	86.77	15.02	17.22	30.3
RE-GCN	66.75	73.65	87.10	24.22	26.79	43.25
LLM-ICL	-	77.2	82.7	-	0.1	0.1
GenTKG	-	**82.1**	86.8	-	16.9	22.8
MESH	**72.81**	80.80	**91.81**	**24.52**	**27.13**	**44.06**

Table 5: Different event types on ICEWS14.

α_1	Historical	Non-Historical
Mean	0.5341	0.4589
Std	0.079	0.094
p-value	< 0.001	

Table 6: T-test for α on ICEWS14

4.6 In-depth Analysis on Event Types

In this section, we conduct three experiments to validate our claim that different event types require distinct types of information, and to explore the optimal number of experts.

Performance on Different Events. As shown in Table 5, we observe distinct performance between graph-based methods and LLM-based methods on different types of events. Graph-based methods like RE-GCN demonstrate strong capability in capturing evolution patterns through their structural modeling, while LLM-based models (e.g., GenTKG) excel particularly at modeling historical events due to their powerful representation learning but show limited generalization to non-historical scenarios. Our proposed method achieves consistent improvements in both scenarios, suggesting its effectiveness in learning specific reasoning patterns for different types of events. This balanced performance can be attributed to our model's ability to leverage both structural patterns and semantic information effectively, bridging the gap between historical/non-historical events.

Statistic Analysis of Prediction Expert. In this part, we present a statistical analysis to demonstrate the ability of the prediction expert to predict different event types with varied patterns. We performed a t-test on α_1, as shown in Table 6. α_1 refers to the weight computed in Equation (7), which is assigned to the historical expert for prediction. As shown in the 'Mean' row, we observe that the mean value of α_1 for historical events is relatively higher than that for non-historical events. With standard deviations calculated from 7,371 samples, we conducted a t-test with the alternative hypothesis that 'The mean

(M, N)	MRR	H@3	H@10
(1,1)	44.36	49.81	64.21
(1,2)	43.94	48.62	63.03
(2,1)	44.08	48.55	63.00
(2,2)	43.91	48.60	62.81
(3,1)	44.19	48.89	62.87
(1,3)	43.91	48.79	63.13
(3,3)	43.81	48.62	62.94

Table 7: Expert configuration tests on ICEWS14.

weight for historical events is greater than that for non-historical events', which was validated with a highly significant p-value ($p << 0.001$).

Sensitivity Analysis on Event-aware Experts Configuration. In this part, we analyze the experiment results of varying the number of historical/non-historical expert modules. As shown in Table 7, the optimal performance is achieved with $(M, N) = (1, 1)$. As the number of experts increases, the prediction performance tends to decrease, indicating that complex combinations of expert modules are not necessary for the TKG Reasoning task. In fact, increasing the number of experts may lead to parameter redundancy and raise the risk of overfitting.

5 Conclusion

In this work, we proposed a Multi-Expert Structural-Semantic Hybrid (MESH) framework. Through the design of expert modules and the event-aware gate function, our model enabled adaptive information fusion based on event types. Extensive experiments on three public datasets demonstrated that our approach consistently outperformed existing methods.

Limitations

The performance of MESH relies on the effectiveness of the underlying structural encoder and semantic encoder. The quality of event representations and the final prediction results are bounded by the capacity of these encoders. In the future, the framework can incorporate more advanced feature encoders and enhance its performance. The modular design allows MESH to benefit from future improvements in representation learning techniques while maintaining its core architecture.

Acknowledgments

This work is in part funded by the National Natural Science Foundation of China (No. 62372364); in part by Shaanxi Provincial Technical Innovation Guidance Plan, China under Grant 2024QCY-KXJ-199; in part by Research Impact Fund (No.R1015-23), Collaborative Research Fund (No.C1043-24GF) and Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program).

References

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

He Chang, Chenchen Ye, Zhulin Tao, Jie Wu, Zhengmao Yang, Yunshan Ma, Xianglin Huang, and Tat-Seng Chua. 2024. A comprehensive evaluation of large language models on temporal event forecasting. *arXiv preprint arXiv:2407.11638*.

Can Chen, Gabriel L Oliveira, Hossein Sharifi-Noghabi, and Tristan Sylvain. 2024. Enhance time series modeling by integrating llm. In *NeurIPS Workshop on Time Series in the Age of Large Models*.

Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.

Julia Gastinger, Christian Meilicke, Federico Errica, Timo Sztyler, Anett Schuelke, and Heiner Stuckenschmidt. 2024. History repeats itself: A baseline for temporal knowledge graph forecasting. *arXiv preprint arXiv:2404.16726*.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6669–6683.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Proceedings of the Web Conference 2018*, pages 1771–1776.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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.

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 *Proceedings of the IJCAI Conference*, 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*, pages 408–417.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024. GenTKG: Generative forecasting on temporal knowledge graph with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4303–4317.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024a. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024b. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114.

Ruilin Luo, Tianle Gu, Haoling Li, Junzhe Li, Zicheng Lin, Jiayi Li, and Yujiu Yang. 2024. Chain of history: Learning and forecasting with llms for temporal knowledge graph completion. *arXiv preprint arXiv:2401.06072*.

Haithem Mezni. 2021. Temporal knowledge graph embedding for effective service recommendation. *IEEE Transactions on Services Computing*, 15(5):3077–3088.

Chong Mo, Ye Wang, Yan Jia, and Qing Liao. 2021. Survey on temporal knowledge graph. In *2021 IEEE Sixth International Conference on Data Science in Cyberspace*, pages 294–300. IEEE.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.

Miao Peng, Ben Liu, Wenjie Xu, Zihao Jiang, Jiahui Zhu, and Min Peng. 2024. Deja vu: Contrastive historical modeling with prefix-tuning for temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1178–1191.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*.

Jiapu Wang, Boyue Wang, Meikang Qiu, et al. 2023a. A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2308.02457*.

Yejing Wang, Zhaocheng Du, Xiangyu Zhao, Bo Chen, Huifeng Guo, Ruiming Tang, and Zhenhua Dong. 2023b. Single-shot feature selection for multi-task recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 341–351.

Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023c. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525*.

Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023d. PLATE: A prompt-enhanced paradigm for multi-scenario recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1498–1507.

Feng Wu, Guoshuai Zhao, Tengjiao Li, Jialie Shen, and Xueming Qian. 2024. Improving conversation recommendation system through personalized preference modeling and knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*.

Sen Wu, Guoshuai Zhao, and Xueming Qian. 2023. Resolving zero-shot and fact-based visual question answering via enhanced fact retrieval. *IEEE Transactions on Multimedia*, 26:1790–1800.

Xiaoying Xie, Biao Gong, Yiliang Lv, Zhen Han, Guoshuai Zhao, and Xueming Qian. 2023. Logic diffusion for knowledge graph reasoning. *arXiv preprint arXiv:2306.03515*.

Derong Xu, Ziheng Zhang, Zhenxi Lin, Xian Wu, Zhihong Zhu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024. Multi-perspective improvement of knowledge graph completion with large language models. *arXiv preprint arXiv:2403.01972*.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023a. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7790–7803.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023b. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2024a. Jasper and stella: distillation of sota embedding models. *arXiv preprint arXiv:2412.19048*.

Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. 2024b. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 893–902.

Zijian Zhang, Xiangyu Zhao, Hao Miao, Chunxu Zhang, Hongwei Zhao, and Junbo Zhang. 2023. Autostl: Automated spatio-temporal multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4902–4910.

Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018a. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pages 95–103.

Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018b. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1040–1048.

Algorithm 1 Training Process of MESH

Require: Training set $\mathcal{D}$, expert weight ω, pre-trained language model LLM
Ensure: Optimized model parameters θ^*

1: **Stage 0:** Pre-training
2: Train structural encoder G on $\mathcal{D}$
3: Freeze parameters of G
4: **Stage 1:** Main Training Process
5: Load G, LLM and obtain entity and relation representation $\boldsymbol{H}_g, \boldsymbol{R}_g, \boldsymbol{H}_{LLM}, \boldsymbol{R}_{LLM}$ (Eq. 1)
6: Initialize model parameters θ
7: **while** Not Convergence **do**
8: **for** event $(s, r, o, t) \in \mathcal{D}$ **do**
9: Obtain $\boldsymbol{h}_g, \boldsymbol{r}_g$ and $\boldsymbol{h}_l, \boldsymbol{r}_l$ (Eq. 2)
10: Calculate structural and semantic query representation $\boldsymbol{q}_g, \boldsymbol{q}_l$ (Eq. 3, 4)
11: Calculate historical identifier $I_t^{s,r}(o)$ (Eq. 15), historical/non-historical query representation $\boldsymbol{q}^{his}, \boldsymbol{q}^{nhis}$ (Eq. 10, 11)
12: Calculate final query representation $\boldsymbol{q}$ (Eq. 8)
13: Make final prediction $\boldsymbol{p}_{s,r,t}$ and expert predictions $\boldsymbol{p}_{s,r,t}^{his}$, $\boldsymbol{p}_{s,r,t}^{nhis}$ (Eq. 9, 12, 13)
14: Compute expert losses $\mathcal{L}_e^{his}$, $\mathcal{L}_e^{nhis}$, prediction loss $\mathcal{L}^m$ (Eq. 16, 17, 18)
15: Get total loss $\mathcal{L}$ (Eq. 19)
16: Update parameters via backpropagation
17: **end for**
18: **end while**
19: **return** Optimized model parameters θ^*

Algorithm 2 Naive Approach

Require: Test set $\mathcal{D}_{test}$, training set $\mathcal{D}_{train}$, rank threshold k
Ensure: Hits@k scores

for each $(s, r, o, t) \in \mathcal{D}_{test}$ **do**
  if $\exists (s, r, *, t') \in \mathcal{D}_{train}$ **then**
    Count frequency of objects interacting with (s, r)
  else
    Count frequency of objects interacting with s
  end if
  $rank \leftarrow$ Position of o in frequency-sorted list
  if $rank \leq k$ **then**
    $hits@k \leftarrow hits@k + 1$
  end if
end for
return $hits@k/|\mathcal{D}_{test}|$

A Optimization Algorithm

To better illustrate the optimization of MESH, we present the complete training process in Algorithm 1, which shows the two-stage training process and the integration of expert modules. First, we train and freeze the parameters of the structural encoder G (lines 2-3). Then, we obtain entity and relation representations from both G and LLM to derive the structural and semantic query representations (lines 5, 9-10). Next, we employ event-aware experts to integrate semantic and structural information, followed by a prediction expert to combine the outputs from event-aware experts, generating integrated query representations and predictions (lines 10-13). Finally, we compute the expert loss to guide the model in learning different patterns for different event types and prediction loss to enhance its reasoning capability, then update the model parameters (lines 14-16).

B Experiment Setups

B.1 Evaluation Metrics

In this section, we introduce the formal definitions of MRR and H@k metrics.

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{20}$$

where $|Q|$ denotes the total number of queries and rank_i represents the rank position of the correct answer for the i-th query.

$$\text{Hit@k} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \phi(\text{rank}_i \leq K) \tag{21}$$

where $\phi(\cdot)$ is an indicator function that returns 1 if the condition is satisfied and 0 otherwise.

B.2 Lightweight Baselines

To investigate the performance of simple baselines in the TKG reasoning task, we propose two lightweight approaches based on dataset statistics or large language model encoding.

The approach "Naive" simply selects the top k entities that have the most interactions with the query or subject as the prediction results, as shown in Algorithm 2. This method is training-free and the execution time is less than 10 seconds. As

Input	MRR	H@3	H@10
Structural	44.36	49.81	64.21
Semantic	44.12	49.31	63.26
Concatenated	44.17	49.40	63.57

Table 8: Gate input analysis on ICEWS14.

Methods	MRR	H@3	H@10
RE-GCN	19.16	20.41	33.10
LLM-MLP	15.71	15.89	26.33
MESH(ours)	19.96	21.43	34.37

Table 9: Experiments on GDELT dataset.

shown in Table 2, it achieves comparable performance to some graph-based methods on both ICEWS14 and ICEWS05-15.

The approach LLM-MLP employs MLP layers to compress the embeddings of entities and relations obtained from LLMs, then utilizes a ConvTransE decoder for prediction, reducing inference costs by more than 90% compared to existing LLM-based approaches. However, it performs well across three datasets compared to other generative LLM-based methods. Specifically, it outperforms ICL across most of metrics.

Experiments of these two approaches reveal that TKG reasoning often follows repetitive patterns. However, it is also important to capture the emergence of new events (non-historical events).

C Additional Experiments

C.1 Ablation on Gating Input

In this section, we conduct experiments of using different inputs (structural, semantic, and concatenated) for the query-motivated gate to compare how different types of information guide the wight allocation and affect prediction results. The experimental results in Table 8 demonstrate that using structural information as input achieves the best performance, because structural information not only contains the information of the query context, but also includes global knowledge of the graph, which can better guide the gate's output.

C.2 Experiments on GDELT Dataset

We have conducted experiments on one more dataset, GDELT, from news domain, which has a finer temporal granularity (15 minutes). It is a global event dataset based on international news reports. We compare our proposed method with graph-based baseline RE-GCN and LLM-based baseline LLM-MLP on the GDELT (Leetaru and Schrodt, 2013) dataset. Experimental results in Table 9 show that our framework, by jointly leveraging structural and semantic information, consistently outperforms approaches that utilize only one type of information.

MESH	MRR	H@3	H@10
ICEWS14, 100%	44.36	49.81	64.21
ICEWS14, 75%	43.91	48.62	63.62
ICEWS14, 50%	42.24	46.6	61.02

Table 10: Experiments on incomplete historical data.

C.3 Experiments on Incomplete Historical Data

To simulate scenarios with incomplete historical data, we randomly removed 25% and 50% of historical event records from the ICEWS14 training set and conducted experiments using our method MESH. The results in Table 10 show that although reducing historical data slightly degrades reasoning performance, our approach remains robust, highlighting both the importance of historical events and our model's resilience to missing information.