LLM-SPM:Semantic-Driven Temporal Knowledge Graph Forecasting via Large Language Models and Pattern Mining

1st Jun Zhu
Big Data Research Center
University of Electronic Science and Technology of China
Chengdu, China
j.zhu@std.uestc.edu.cn

2nd Tianxiang He
Chengdu Union Big Data Tech Inc.
Chengdu, China
hetianxiang@unionbigdata.com

3rd Yan Fu
Big Data Research Center
University of Electronic Science and Technology of China
Chengdu Union Big Data Tech Inc.
Chengdu, China
fuyan@uestc.edu.cn

4th Junlin Zhou
Big Data Research Center
University of Electronic Science and Technology of China
Chengdu Union Big Data Tech Inc.
Chengdu, China
jlzhou@uestc.edu.cn

5th Duanbing Chen*
Big Data Research Center
University of Electronic Science and Technology of China
Chengdu Union Big Data Tech Inc.
Chengdu, China
dbchen@uestc.edu.cn

***Abstract*—Temporal Knowledge Graph (TKG) forecasting is a critical task in the field of Knowledge Graphs (KGs), aiming to predict events that may occur in the future. However, only structured information or timestamps are considered in the process of reasoning, and the semantics of entities and relations are ignored in the current models. This semantic information includes potential evolutionary laws, which are beneficial for reasoning for more accurate results. To address this limitation, we propose a novel framework driven by Large Language Models (LLMs) for enhancing TKG forecasting by Semantic Pattern Mining, named LLM-SPM. By leveraging the natural language understanding and zero-shot learning capabilities of LLMs, entity type information in TKGs is automatically extracted. Next, based on the inferred entity types and relations, semantic pattern triples are constructed. Finally, historical facts in the TKG are reweighted or filtered through a comprehensive scoring mechanism that integrates multiple factors, including temporal decay, pattern relevance, historical recurrence, and an entity specificity measure, enabling the model to focus on facts that are more relevant and semantically coherent. Through this approach, deep semantic knowledge is effectively incorporated into the prediction process, substantially improving reasoning accuracy in various scenarios. Experiments on benchmark datasets demonstrate that the LLM-SPM framework outperforms both previous methods and standard LLMs.**

***Index Terms*—temporal knowledge graph, large language model, pattern mining**

I. Introduction

A Temporal Knowledge Graph (TKG) is a dynamic graph structure that represents knowledge as quadruples *(subject entity, relation, object entity, timestamp)*. Based on the Knowledge Graph (KG), not only are static connections between entities established, but also the process of relation evolution over time is captured. The goal of forecasting over TKG is to predict facts that may emerge at future based on historical facts [1]. This task provides a foundational data infrastructure for dynamic event prediction across diverse domains, including financial market forecasting (e.g., predicting stock-merger relationships), social network evolution analysis (e.g., anticipating collaboration patterns), and temporal recommendation systems (e.g., forecasting user-item interactions).

In recent years, embedding-based approaches and Graph Neural Network (GNN)-based approaches [2], [3] have achieved significant progress in forecasting tasks by learning low-dimensional representations of entities and relations to capture the structural and temporal evolution patterns of the graph. However, these methods have a critical limitation: these approaches effectively model *who connects to whom* and *when*, and they pay little attention to the semantic motivations behind *why such relationships arise*. For example, when forecasting an event like *(Company A, acquire, ?, 2025-12-01)*, if the model recognizes that company entities historically exhibit frequent acquisition relationships, the confidence in such predictions should increase. This semantic pattern information—rooted in entity types—is often overlooked in traditional approaches.

The emergence of large language models (LLMs) offers new potential to address this limitation [4]. Fine-tuned on vast text corpora, LLMs encode extensive knowledge of the world and robust semantic understanding capabilities, enabling them to identify entities [5], comprehend concepts, and perform complex logical reasoning. In recent years, several LLM-

*Corresponding author.

based reasoning approaches have emerged in the temporal knowledge graph domain, aiming to enhance model generalization across diverse scenarios. However, most approaches rely on simplistic criteria—such as chronological filtering—to select relevant historical information, then construct prompts to fine-tune LLMs for prediction tasks, without fully leveraging their capabilities in semantic understanding and multi-step reasoning.

Compared to conventional embedding-based or neural network-based approaches, LLM-driven methods explicitly incorporate the semantic information of entities and relations into their reasoning pipelines. However, a critical challenge persists: most entity names inherently lack semantic transparency—for example, personal names (e.g., *John Smith*) or alphanumeric identifiers (e.g., *Company_X*) carry no intrinsic meaning for models to extract semantic rules. When such opaque identifiers are directly fed into LLMs for single-step inference, the models struggle to discover meaningful semantic patterns that could guide the filtering of relevant historical facts.

To address the limitations of conventional TKG forecasting approaches, we propose a novel framework named **LLM-SPM** (Large Language Model-driven Semantic Pattern Mining) that leverages LLMs as *semantic preprocessors* rather than direct end-to-end predictors [1]. The key contributions are:

- Utilizing LLMs' zero-shot capabilities to infer hierarchical entity types (e.g., *Person*, *Organization*, *Location*) from raw entity names or descriptions without manual annotation. This is particularly critical for datasets where explicit type metadata is absent, enabling the extraction of semantic hierarchies directly from unstructured textual sources.
- Generating abstract schema-level patterns by analyzing co-occurrence relationships between entity types and relations. These patterns capture generalized semantic rules that govern interactions within the temporal knowledge graph, providing a structured abstraction of the underlying data.
- Developing a semantic-aware filtering mechanism that dynamically prioritizes historical facts based on their relevance to the current prediction task. This mechanism integrates temporal dynamics, pattern coherence, historical recurrence, and entity-specific characteristics to refine the input data for downstream models.
- Demonstrating consistent improvements in forecasting accuracy and robustness across standard TKG datasets. By incorporating semantically enriched inputs, the framework enhances the ability of existing models to generalize in data-scarce scenarios and complex reasoning tasks.

The experimental results of this work demonstrate that systematic integration of LLM-extracted semantic knowledge can effectively overcome critical limitations in existing TKG forecasting methods, particularly in environments with sparse data or requiring multi-step logical inference.

[1]The source code is available at https://github.com/jzhu000/LLM-SPM.git.

II. Related Works

In the field of temporal knowledge graph (TKG) forecasting, traditional approaches—such as those based on graph neural networks (GNNs), embedding models, and symbolic rules—have played a pivotal role. These methods have demonstrated strong performance on TKG benchmarks and established diverse theoretical foundations for the task of temporal link prediction. However, their generalization capabilities across different domains or scenarios remain limited due to reliance on dataset-specific patterns and structural assumptions.

Recently, the advent of large language models (LLMs) has introduced new opportunities in this field. By integrating LLMs with traditional TKG techniques, recent work has shown that such hybrid approaches can significantly enhance model generalization, leveraging the rich semantic priors and in-context reasoning abilities of LLMs to overcome the brittleness of purely structure-based methods.

The following sections review representative works in two categories: (1) traditional TKG forecasting methods and (2) LLM-driven approaches.

A. Traditional Methods

Most traditional methods mainly utilize GNNs to capture interactions hidden in the graph structure, or reasons based on a rule system. At the same time, embeddings play a role in representing entities and relations over time. RETIA [6] aggregates interactions between entities and relations of temporal subgraphs to reason future events relying on GNNs. LogE-Net [7] introduces logic rules based on RE-GCN, utilizing rules to preprocess data for logical reasoning. TECHS [8] integrates a temporal graph encoder with a logical decoder for explainable extrapolation. L2TKG [9] mines intra- and inter-time latent relations via a structural encoder. DGExplainer [10] extends this by providing explanations for dynamic GNN predictions via relevance back-propagation, though it focuses more on interpretability than direct forecasting, with applications in link prediction on traffic datasets. CENET [11] uses historical contrastive learning to distinguish dependencies, with a mask-based inference for future predictions. TRKG [12] extends TLogic [13] with time-relaxed cyclic and acyclic rules.

B. LLM-driven Approaches

LLM-driven methods usually rely on in-context learning, sometimes combined with parameter-efficient fine-tuning, to adapt to the task of TKG forecasting. These approaches leverage LLMs' semantic understanding to reason over temporal facts without explicit graph modules, often requiring fewer resources than traditional methods. They excel in zero-/few-shot settings but face challenges in aligning textual prompts with graph structures. GPT-NeoX [14] demonstrates that LLMs can perform TKG prediction via ICL alone, achieving near-supervised performance on ICEWS datasets even when entities/relations are anonymized to numbers. GenTKG [15] combines few-shot instruction tuning with a temporal logical rule retrieval (TLR) mechanism to generate future facts, excelling in cross-domain and unseen settings. Chain of history [4] uses

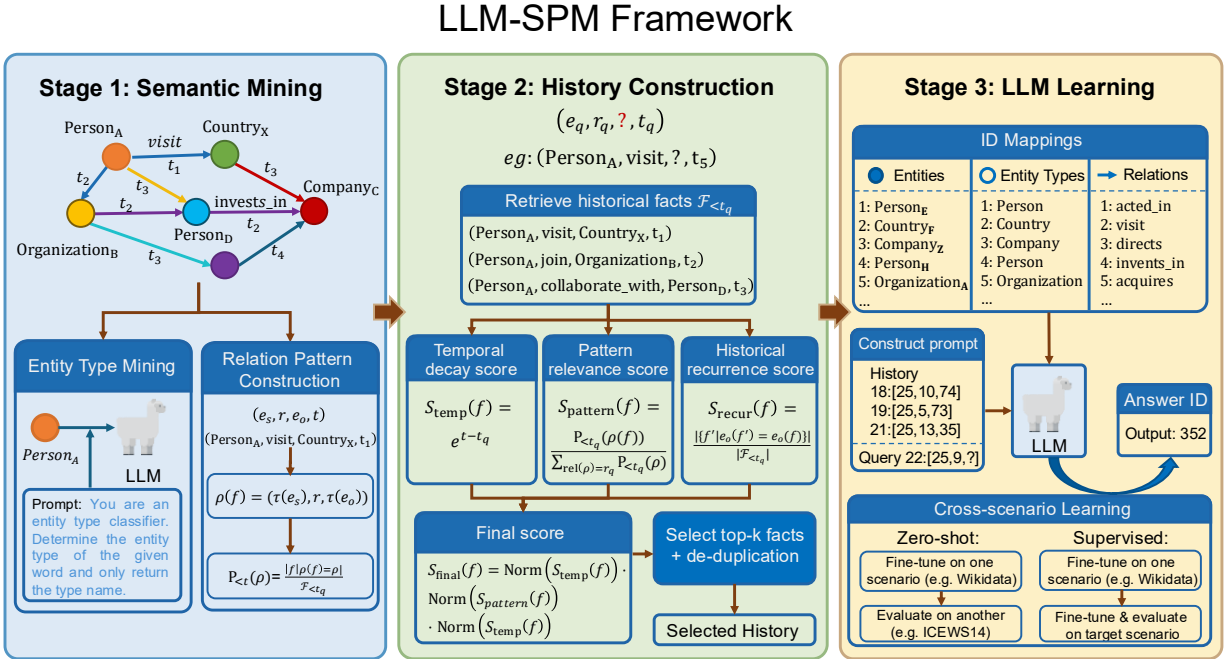

Fig. 1. The framework of LLM-SPM.

low-rank adaptation (LoRA) [16] fine-tuning on LLMs with structure-augmented histories and reverse chains to counter the reversal curse. LLM-DA [17] leverages LLM-generated insights/rules for bidirectional temporal adaptation, addressing dynamic graph challenges in reasoning/forecasting. LLM-DR [18] combines LLM prompting with a diffusion process to produce high-quality, temporally aware rules, handling redundancy and improving explainable extrapolation. G2S [19] introduces a general-to-specific learning framework leveraging LLMs' prior knowledge while adapting to TKG rules.

III. Problem Definition

A TKG $\mathcal{G}$ can be defined as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$, where $\mathcal{E}$ denotes entities, $\mathcal{R}$ denotes relations, $\mathcal{T}$ denotes timestamps, and $\mathcal{F}$ denotes facts. Each fact can be formalized as a quadruple $(e_s, r, e_o, t) \in \mathcal{F}$, where subject and object entities $e_s, e_o \in \mathcal{E}$, relation $r \in \mathcal{R}$, timestamp $t \in \mathcal{T}$. For the forecasting task over TKG, the main goal is predicting the missing entity in the query $f_q = (e_{s,q}, r_q, ?, t_q)$ or $(?, r_q, e_{o,q}, t_q)$ based on the history $\mathcal{F}_{<t_q}$. In this paper, the task of model $\mathcal{M}$ is compute the score of each candidate entity $Score(e_{\cdot,q}) = \mathcal{M}(f_q, \mathcal{F}_{<t_q})$.

IV. Methodology

A. Framework Construction

As shown in Figure 1, the proposed methodology can be divided into three stages:

- **Stage 1: Semantic Mining.** The semantic mining stage aims to mine the potential semantic information from original TKG elements, including mining types of entities by LLMs and capturing potential patterns from TKG.
- **Stage 2: History Construction.** History Construction is based on the semantic rules mining in the previous stage. Various rating mechanisms are utilized to evaluate each historical fact. Then top-k important historical facts are chosen as the basis for reasoning by LLMs.
- **Stage 3: LLM Learning.** Building on the previous stages, a prompt is constructed using selected historical facts to guide the LLM's reasoning. Two distinct learning paradigms are then employed to enhance and evaluate the model's generalization capability.

B. Semantic Mining

1) Entity Type Mining: For each entity in TKG, the name is the unique information in the original data. The ID is usually used as the identifier in most previous methods, and more potential semantic information is ignored. Moreover, many entity names—particularly personal names—serve only as identifiers and carry little to no semantic meaning. Consequently, even when original names are used as input, they often provide limited utility for event prediction. To address this, we leverage large language models (LLMs) to infer high-level entity types and exploit the inherent relational structure among these types to guide reasoning. For instance, given a query *(Person, acted_in, ?, 2015-01-03)*, the model can infer that the answer should be of type Movie rather than Company, based on the semantic compatibility between the head entity type *(Person)* and the relation *(acted_in)*. For entity e, the entity type can be obtained by LLM:

$$\tau(e) = LLM(e, prompt_{type}) \tag{1}$$

2) Relation Pattern Construction: Once entity types are obtained, they can be leveraged in two complementary ways to guide LLM reasoning. First, type-related commonsense knowledge—acquired by the LLM during pretraining—can

inform predictions, enabling the model to draw on its broad understanding of real-world semantics. Second, relational patterns composed of *(subject-type, relation, object-type)* provide structured semantic cues that further constrain and direct inference.

Notably, a single relation may correspond to multiple valid patterns. For example, the relation visit can appear in patterns such as (Person, visit, Person) or (Person, visit, Country). Patterns that have occurred frequently in historical data are more likely to recur in the future. By counting the number of observed facts associated with each pattern, one can estimate the empirical occurrence probability of different relational patterns for a given relation, thereby offering a data-driven prior for future predictions. For a fact $f = (e_s, e_r, e_o, t)$, its relation pattern can be defined as:

$$\rho(f) = (\tau(e_s), r, \tau(e_o)) \tag{2}$$

For the pattern ρ, the empirical occurrence probability can be computed by:

$$\mathrm{P}_{<t}(\rho) = \frac{|\{f|\rho(f) = \rho, f \in \mathcal{F}_{<t}\}|}{|\mathcal{F}_{<t}|} \tag{3}$$

where $\mathrm{P}_{<t}$ denotes the empirical occurrence probability of facts related to events that happened prior to time t.

C. History Construction

For each query $(e_s, r_q, ?, t)$, we can find related historical fact set $\{(e_q, r_q, e_{q,o}, t) \in \mathcal{F}_{<t_q}\}$, then guide LLM to reason answer based on the history. However, the context window of large language models (LLMs) is inherently limited, necessitating the selection of the most semantically meaningful historical facts. To address this constraint, we propose a scoring mechanism that integrates temporal decay, pattern relevance, and entity specificity to evaluate each historical fact. By ranking the facts according to their computed scores and retaining the top-k instances, we construct a compact yet semantically enriched historical context for LLM reasoning. This prioritization ensures that the model focuses on temporally recent, structurally coherent, and entity-specifically significant facts when making predictions.

Temporal decay score. The relevance of historical facts diminishes as their temporal distance from the target prediction timestamp increases. For a historical fact $(e_q, r, e_{q,o}, t)$, temporal decay score is computed as:

$$S_{\text{temp}}(f) = e^{t - t_q} \tag{4}$$

where the exponential function ensures that historical facts with timestamps closer to the query time receive higher scores. It also avoids the risk of producing negative scores that would occur from directly subtracting time differences.

Pattern relevance score. Building on the empirical occurrence probability of a pattern computed earlier, the pattern relevance of a historical fact is determined by its normalized historical frequency:

$$S_{\text{pattern}}(f) = \frac{\mathrm{P}_{<t_q}(\rho(f))}{\sum_{\text{rel}(\rho)=r_q} \mathrm{P}_{<t_q}(\rho)} \tag{5}$$

where $\rho(f)$ represents the relation pattern of fact f, and $\mathrm{P}_{<t_q}(\cdot)$ denotes the empirical occurrence probability during the history before t_q, $\text{rel}(\rho)$ denotes the relation of the pattern ρ. This score quantifies how well a fact aligns with dominant semantic patterns.

Historical recurrence score. Entities that frequently interacted with the query entity e_q in the past are more likely to reappear in future events. We formalize this as:

$$S_{\text{recur}}(f) = \frac{|\{f'|e_o(f') = e_o(f)\}|}{|\mathcal{F}_{<t_q}|} \tag{6}$$

where $e_o(f)$ denotes the object entity of the fact f, $|\mathcal{F}_{<t_q}|$ denotes the number of facts during the history $< t_q$.

Final fact score The final score of fact f is computed with three scores behind:

$$\begin{aligned} S_{\text{final}}(f) = \text{Norm}(S_{\text{temp}}(f)) \cdot \text{Norm}(S_{\text{pattern}}(f)) \\ \cdot \text{Norm}(S_{\text{recur}}(f)) \end{aligned} \tag{7}$$

To avoid influence between different scores, each score is normalized before computing the final score.

Entity specificity consideration. At the same time, we introduce a de-duplication mechanism to avoid overfitting to frequently occurring entities. After sorting facts by the final score, we filter out subsequent facts involving already-included entities, ensuring diversity in the final context window. This design prevents duplicate entities from monopolizing the LLM's attention during multi-step reasoning tasks.

D. Cross-Scenario Learning

The prompt for finetuning LLMs can be constructed based on the captured history. However, directly prompting LLMs to generate answer entities based on historical facts may result in letter or word omissions. To address this, following previous approaches [19], we represent entities, relations, and timestamps using IDs, and introduce ID mappings for entities, relations, and entity types. For timestamps, we omit explicit mapping and instead use monotonically increasing IDs that evolve uniformly with historical progression. This allows both historical and query facts to be expressed as IDs, significantly reducing the input context length required. Furthermore, representing elements via IDs enhances LLM generalization across diverse scenarios, as the model learns to predict answer IDs based on contextual mappings rather than surface-level entity names.

To evaluate the model's generalization capabilities across diverse scenarios, we design two learning paradigms: zero-shot learning and supervised learning. In the zero-shot setting, the model is fine-tuned and evaluated on datasets from distinct scenarios (e.g., fine-tuning on Wikidata and evaluating on ICEWS14), specifically targeting the assessment of cross-scenario generalization. In the supervised setting, the model first undergoes pre-training on external scenario datasets, followed by fine-tuning and evaluation on the target dataset. This protocol ensures a comprehensive analysis of both domain adaptation and task-specific performance.

TABLE I
STATISTICS OF BENCHMARK DATASETS

	ICEWS14	ICEWS18	YAGO	WIKI	GDELT
$\lvert\mathcal{E}\rvert$	7,128	23,033	10,623	12,513	7,691
$\lvert\mathcal{R}\rvert$	230	256	10	24	240
$\lvert\mathcal{T}\rvert$	365	304	189	232	2,751
$\lvert\mathcal{F}_{train}\rvert$	74,845	373,018	161,540	539,286	1,734,399
$\lvert\mathcal{F}_{valid}\rvert$	8,514	45,995	19,523	67,538	238,765
$\lvert\mathcal{F}_{test}\rvert$	7,371	49,545	20,026	63,110	305,241

TABLE II
DETAILS FOR EXPERIMENTS ON ALL DATASETS

Hyperparameter	**Value**
Number of historical facts	64
Context length of prompts	1536
Learning rate	1×10^{-4}
Batch size	32
Number of epochs	2

V. EXPERIMENTS

A. Datasets

In this work, five widely adopted TKG datasets are used for fine-tuning and evaluation, including ICEWS14 [20], ICEWS18 [21], WIKI [22], YAGO [23], and GDELT [24]. Statistics of them are shown in Table I, where $|\mathcal{E}|$ denotes the number of entities, $|\mathcal{R}|$ denotes the number of relations, $|\mathcal{T}|$ denotes the number of timestamps, $|\mathcal{F}_{train}|$ denotes the number of fine-tuning facts, $|\mathcal{F}_{valid}|$ denotes the number of valid facts, and $|\mathcal{F}_{test}|$ denotes the number of test facts.

B. Metrics

Evaluation of the proposed approach relies on the standard ranking-based metrics Hits@1, Hits@3, and Hits@10, which quantify the fraction of test queries where the true answer appears among the top 1, 3, or 10 predicted entities, respectively. Mean Reciprocal Rank (MRR), despite its prevalence in earlier knowledge graph reasoning studies, is not adopted here due to its incompatibility with large language models (LLMs): unlike traditional link prediction models, LLMs do not produce a full ranked list over the entire entity vocabulary, making MRR undefined or unreliable.

C. Baselines

Some traditional methods and LLM-based methods are chosen as baselines for comparison with the proposed methods. Traditional methods include GNN-based and logic rule-based RE-GCN [25], xERTE [26], TANGO [27], TITer [28], and TLogic [13]. LLM-based in-context learning methods mainly include Llama2-7B [29], Llama3-8B [30], and GPT-NeoX-20B [14]. In addition, some rule-based models with LLMs are also considered, including GenTKG [15], and G2S [19].

G2S serves as the primary baseline model selected for comparative evaluation in this work. To ensure a fair comparison, we adopted an identical data utilization strategy: selecting 100,000 fine-tuning instances from the GDELT dataset and 30,000 instances from the WIKI dataset as source scenario data for model fine-tuning. Subsequently, the model was evaluated on three target datasets—ICEWS14, ICEWS18, and YAGO—through two distinct protocols: (1) supervised learning, where the model undergoes fine-tuning on the target dataset before evaluation, and (2) zero-shot learning, where the model is evaluated directly without prior exposure to the target dataset.

D. Experiment Details

In the learning process, datasets are classified into two groups: source scenario group, including GDELT and WIKI, and target scenario group, including ICEWS14, ICEWS18, and YAGO. For the **zero-shot setting**, the model is only fine-tuned with datasets from the source scenario group. For the **supervised setting**, the model is first fine-tuned with fine-tuning samples from the source scenario group, then fine-tuned and evaluated with each dataset from the target scenario group.

For each learning setting, the parameters are configured as listed in Table II. In this paper, LlamaFactory [31] is utilized as a tool to fine-tune and evaluate LLMs in our project, and Llama-3.1-8B-Instruct [30] is chosen as the base LLM for LoRA finetuning on datasets. The experiments are run on an NVIDIA H200 GPU.

E. Results

Under the supervised learning setting, the large language model is first fine-tuned on source-scenario data and then fine-tuned and evaluated separately on each target-scenario dataset. In contrast, under the zero-shot learning setting, the model is fine-tuned only on the source-scenario data and directly evaluated on each target-scenario dataset without any further fine-tuning. The evaluation results are presented in the table III [2], where entries marked with **ZS** indicate performance obtained under the zero-shot condition. For models with ICL, they are all under the zero-shot setting.

Under the supervised learning setting, our proposed model outperforms all conventional approaches, demonstrating the strong reasoning capability of large language models (LLMs) in the domain of temporal knowledge graphs (TKGs). Moreover, compared to all existing LLM-based in-context learning methods, our model achieves substantial improvements, validating the effectiveness of our key strategies—semantic pattern mining, historical fact reconstruction, and multi-stage learning.

Notably, our proposed model (LLM-SPM) achieves superior performance over previous methods across most evaluation metrics in both supervised and zero-shot settings, with only minor exceptions on a few isolated cases (e.g., slightly lower Hits@10 on ICEWS18 compared to G2S under the supervised setting, and on YAGO compared to certain ICL baselines).

[2]The results of G2S were reproduced using the official code available at https://github.com/waltbai/G2S-TKG-forecasting. We also attempted to reproduce the results of GenTKG using the code at https://github.com/mayhugotong/GenTKG, but encountered issues that prevented successful execution (possibly due to environment or dependency conflicts). Therefore, we report the GenTKG results directly as published in the original paper.

TABLE III
RESULTS OF TKG FORECASTING ON BENCHMARK DATASETS

Model	ICEWS14			ICEWS18			YAGO		
	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10
RE-GCN	31.30	47.30	62.60	22.30	36.70	52.50	78.70	84.20	88.40
xERTE	33.00	45.40	57.00	20.90	33.50	46.20	84.20	90.20	91.20
TANGO	27.20	40.80	55.00	19.10	31.80	46.20	59.00	64.60	67.70
TITer	31.90	45.40	57.50	21.20	32.50	43.90	84.50	90.80	91.20
TLogic	33.20	47.60	60.20	20.40	33.60	48.00	74.00	78.90	79.10
llama2-ICL	25.80	43.00	51.00	13.50	27.60	32.60	67.70	79.00	81.80
llama3-ICL	31.90	42.40	44.10	18.10	27.20	28.80	56.90	57.90	57.90
GPT-NeoX-ICL	32.40	46.00	56.50	19.20	31.30	41.40	78.70	89.20	**92.60**
G2S(ZS)	31.57	43.70	54.14	19.72	30.65	42.51	80.24	88.25	88.82
GenTKG	36.85	47.95	53.50	24.25	**37.25**	42.10	79.15	83.00	84.25
G2S	37.57	53.87	65.76	22.79	35.01	**45.08**	88.04	89.70	89.93
LLM-SPM(ZS)	31.73	44.49	55.73	19.50	31.78	43.00	82.44	89.61	90.92
LLM-SPM	**38.59**	**56.38**	**67.89**	**24.46**	37.17	44.58	**89.05**	**89.98**	91.77

TABLE IV
RESULTS OF DIFFERENT RATING STRATEGIES

Model	ICEWS14		
	H@1	H@3	H@10
$w/o.S_{\text{temp}}$	37.85	52.95	66.50
$w/o.S_{\text{pattern}}$	35.85	50.95	60.05
$w/o.S_{\text{recur}}$	34.15	51.15	59.70
$w/o.ent\_spec$	33.85	49.65	58.10
$w/.S_{\text{rel}}$	37.89	53.95	64.50
LLM-SPM	38.59	56.38	67.89

This advantage stems from our approach's ability to uncover latent semantic structures within TKGs using LLMs and to construct high-level relational patterns grounded in these semantics—capabilities largely absent in prior methods. In particular, compared to G2S, we employ a more sophisticated and fine-grained scoring mechanism for historical facts, enabling the selection of a richer and more relevant set of contextual evidence, which directly contributes to higher predictive performance.

While our model exhibits slightly lower performance on a small number of isolated metrics (primarily Hits@10 on one or two datasets under specific settings), these exceptions do not undermine the overall superiority of our framework. It consistently demonstrates state-of-the-art or leading performance across the majority of Hits@1, Hits@3, and Hits@10 metrics on all three benchmark datasets (ICEWS14, ICEWS18, and YAGO), outperforming previous LLM-based (including GenTKG, G2S, and ICL methods) and traditional approaches in most cases. This highlights the robustness and effectiveness of our semantic pattern mining, historical fact reconstruction, and multi-stage learning strategies in diverse temporal knowledge graph scenarios.

F. Ablation Study

1) Comparison of Rating Strategies: On the ICEWS14 dataset, we evaluate the effectiveness of different scoring strategies. As described in Section IV-C, the proposed method integrates four scoring components. In Table IV:

- $w/o.S_{\text{temp}}$ denotes the absence of the temporal decay score for LLM-SPM,
- $w/o.S_{\text{pattern}}$ denotes LLM-SPM excluding the pattern relevance score,
- $w/o.S_{\text{recur}}$ denotes removing the historical recurrence score from LLM-SPM,
- $w/o.ent\_spec$ denotes LLM-SPM without the entity specificity mechanism mentioned in Section IV-C.

The results demonstrate that each strategy contributes meaningfully to the overall performance. From a temporal perspective, historical facts closer to the target prediction timestamp exhibit higher relevance—this is validated by $w/o.S_{\text{temp}}$ against the full method. The pattern relevance scoring (S_{pattern}) mechanism evaluates cross-type interactions through historical relational patterns, where frequently co-occurring entity types are more likely to reappear in future interactions. This aligns with real-world observations: entities with stable interaction frequencies (e.g., recurring diplomatic meetings) tend to maintain predictable relational dynamics.

Historical recurrence score (S_{recur}) follows an intuitive principle: entities that repeatedly interacted with the target entity in the past show stronger correlations, reflecting the empirical regularity that frequent historical interactions imply future connectivity. ent_spec plays an equally critical role by preventing redundancy-induced bias. Without ent_spec, strategies pattern relevance score (S_{pattern}) and historical recurrence score (S_{recur}) risk overemphasizing dominant entities, leading to narrowed candidate pools—a limitation effectively mitigated by our scoring framework.

Additionally, we also explored an alternative scoring strategy that assesses the relevance between historical facts and the target prediction based on the frequency of the occurrence of historical relationship, which is then incorporated into the

TABLE V
RESULTS OF LEARNING WITH DIFFERENT DATA CONFIGURATIONS

Model	ICEWS14		
	H@1	H@3	H@10
$w/o.Ent$	36.23	52.58	64.63
$w/o.Ent\_type$	37.02	53.33	66.59
$w/o.Rel$	37.93	55.59	67.10
$w/o.Ent\_type\&Ent$	35.13	51.53	62.92
$w/o.Map$	34.85	50.95	60.05
w/o.GDELT	36.85	52.53	64.10
w/o.WIKI	37.05	54.75	65.50
w/o.GDELT&WIKI	35.87	52.04	62.50
LLM-SPM(full)	38.59	56.38	67.89

final fact score. Similar to the historical recurrence score, this relation frequency score is computed as follows:

$$S_{\text{rel}}(f) = \frac{|\{f'|r(f') = r(f)\}|}{|\mathcal{F}_{<t_q}|} \tag{8}$$

where $r(f)$ denotes the relation of the fact f. The results, including the relation frequency score $w/.S_{\text{rel}}$, are shown in the Table IV. In contrast to other strategies, incorporating this component yields a negative gain. This indicates that facts involving relations that frequently occurred in the past do not necessarily exhibit high relevance to future events. Indeed, such an assumption lacks strong logical grounding in real-world scenarios—past repetition of a relation alone does not reliably imply its recurrence in the future.

2) Impact of Data Configurations: Furthermore, we evaluated the impact of different data configurations on the learning effectiveness of large language models (LLMs) under both supervised and zero-shot settings. We considered the following data configurations.

- $w/o.Ent$ denotes missing entity mappings during fine-tuning on the target scenario data,
- $w/o.Ent\_type$ denotes missing entity type mappings during fine-tuning on the target scenario data,
- $w/o.Rel$ denotes missing relation mappings during fine-tuning on the target scenario data,
- $w/o.Ent\_type\&Ent$ denotes missing entity and entity type mappings during fine-tuning on the target scenario data,
- $w/o.Map$ denotes missing all mappings during fine-tuning on the target scenario data,
- w/o.GDELT denotes missing GDELT data during fine-tuning on the source scenario data,
- w/o.WIKI denotes missing WIKI data during fine-tuning on the source scenario data,
- w/o.GDELT&WIKI denotes missing GDELT and WIKI data during fine-tuning on the source scenario data,

We fine-tuned the models with these incomplete data and then performed evaluations on the ICEWS dataset. The evaluation results are summarized in the Table V. When the fine-tuning data lack entity, relation, or entity type mappings, there is a noticeable degradation in performance. Specifically, the absence of both entity and entity type mappings leads to a more significant performance drop. This is because without entity and entity type mappings, IDs cannot effectively link to entity names and types, hindering the model from making accurate inferences based on entity semantics. However, when only entity type mappings are missing, the model can still leverage entity name mappings for reasoning, though the precision decreases due to the many-to-one relationship between entities and their types.

In the zero-shot fine-tuning phase, missing data also negatively impacts performance. When GDELT data are absent, the model's performance deteriorates more significantly compared to when WIKI is missing. This is primarily because, during zero-shot learning, the GDELT dataset is much larger and thus more influential than the relatively smaller WIKI dataset. Consequently, the absence of WIKI has a lesser effect on overall model performance.

VI. CONCLUSION

This work investigates the critical role of semantic information mining in TKG forecasting. First, we utilize LLMs to extract latent entity type semantics from raw entity names, thereby endowing the model with explicit, structured semantic knowledge. Building upon the inferred entity types, we uncover latent relational patterns from historical facts and design a multi-criteria historical fact filtering mechanism that integrates temporal decay, pattern consistency, and historical recurrence principles. This mechanism assigns composite relevance scores to historical facts, enabling the selection of those most pertinent to the target prediction. Using the filtered facts, we construct structured prompts for fine-tuning LLMs. Moreover, we decouple the fine-tuning process into source-scenario pre-training and target-scenario adaptation, establishing two distinct learning paradigms: supervised learning and zero-shot learning. Experimental results under both settings demonstrate that the proposed framework significantly enhances the generalization capability of LLMs in TKG forecasting. Our findings validate the effectiveness of semantic-aware pattern mining and rule-guided context construction in temporal reasoning tasks, offering an efficient and principled technical framework for LLM-based temporal knowledge graph inference.

ACKNOWLEDGMENT

This work was partially supported by the Innovation Research Group Project of the Natural Science Foundation of Sichuan with Grant No. 2024NSFTD0050, and the Major Program of the National Natural Science Foundation of China with Grant No. T2293771.

REFERENCES

[1] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "Temporal knowledge graph reasoning based on evolutional representation learning," *SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 408–417, 2021.

[2] I. Balazevic, C. Allen, and T. M. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5184–5193.

[3] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs," in *Proceedings of the 34th International Conference on Machine Learning*. Sydney, NSW, Australia: PMLR, 2017, pp. 3462–3471.

[4] Y. Xia, D. Wang, Q. Liu, L. Wang, S. Wu, and X.-Y. Zhang, "Chain-of-history reasoning for temporal knowledge graph forecasting," in *Findings of the Association for Computational Linguistics: ACL*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 16 144–16 159.

[5] R. Aly, A. Vlachos, and R. McDonald, "Leveraging type descriptions for zero-shot named entity recognition and classification," 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)*. Virtual Event: Association for Computational Linguistics, 2021, pp. 1516–1528.

[6] K. Liu, F. Zhao, G. Xu, X. Wang, and H. Jin, "RETIA: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation," in *IEEE 39th International Conference on Data Engineering (ICDE)*. Anaheim, CA, USA: IEEE, 2023, pp. 1761–1774.

[7] Y. Liu, Y. Mo, Z. Chen, and H. Liu, "Loge-net: Logic evolution network for temporal knowledge graph forecasting," in *Artificial Neural Networks and Machine Learning – International Conference on Artificial Neural Networks*. Heraklion, Crete, Greece: Springer, 2023, pp. 472–485.

[8] Q. Lin, J. Liu, R. Mao, F. Xu, and E. Cambria, "TECHS: temporal logical graph networks for explainable extrapolation reasoning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 1281–1293.

[9] M. Zhang, Y. Xia, Q. Liu, S. Wu, and L. Wang, "Learning latent relations for temporal knowledge graph reasoning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 12 617–12 631.

[10] Y. Liu, J. Xie, and Y. Shen, "Dgexplainer: Explaining dynamic graph neural networks via relevance back-propagation," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*. Montreal, Canada: ijcai.org, 2025, pp. 3135–3143.

[11] Y. Xu, J. Ou, H. Xu, and L. Fu, "Temporal knowledge graph reasoning with historical contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*. Washington, DC, USA: AAAI Press, 2023, pp. 4765–4773.

[12] R. U. Kiran, A. Maharana, and K. R. Polepalli, "A novel explainable link forecasting framework for temporal knowledge graphs using time-relaxed cyclic and acyclic rules," in *Advances in Knowledge Discovery and Data Mining – Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Osaka, Japan: Springer, 2023, pp. 264–275.

[13] Y. Liu, Y. Ma, M. Hildebrandt, M. Joblin, and V. Tresp, "Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*. Virtual Event: AAAI Press, 2022, pp. 4120–4127.

[14] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, and H. He, "GPT-NeoX-20B: An open-source autoregressive language model," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*. Virtual: Association for Computational Linguistics, May 2022, pp. 95–136.

[15] R. Liao, X. Jia, Y. Li, Y. Ma, and V. Tresp, "Gentkg: Generative forecasting on temporal knowledge graph with large language models," in *Findings of the Association for Computational Linguistics: NAACL*. Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 4303–4317.

[16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*. Virtual Event: OpenReview.net, 2022, pp. 1–13.

[17] J. Wang, K. Sun, L. Luo, W. Wei, Y. Hu, A. W. Liew, S. Pan, and B. Yin, "Large language models-guided dynamic adaptation for temporal knowledge graph reasoning," in *Advances in Neural Information Processing Systems*. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, Inc., 2024, pp. 8384–8410.

[18] K. Chen, X. Song, Y. Wang, L. Gao, A. Li, X. Zhao, B. Zhou, and Y. Xie, "LLM-DR: A novel llm-aided diffusion model for rule generation on temporal knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*. Philadelphia, PA, USA: AAAI Press, 2025, pp. 11 481–11 489.

[19] L. Bai, Z. Li, X. Jin, J. Guo, X. Cheng, and T. Chua, "G2S: A general-to-specific learning framework for temporal knowledge graph forecasting with large language models," in *Findings of the Association for Computational Linguistics: ACL*. Vienna, Austria: Association for Computational Linguistics, 2025, pp. 20 927–20 938.

[20] A. Garcia-Duran, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4816–4821.

[21] W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inference over temporal knowledge graphs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 6669–6683.

[22] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *Companion Proceedings of The Web Conference*. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 1771–1776.

[23] F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *Conference on Innovative Data Systems Research*. Asilomar, CA, USA: CIDR Conference, 2014, pp. 1–11.

[24] K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in *ISA annual convention*, vol. 2, 2013, pp. 1–49.

[25] Q. Ma, X. Zhang, Z. Ding, C. Gao, W. Shang, Q. Nong, Y. Ma, and Z. Jin, "Temporal knowledge graph reasoning based on evolutional representation and contrastive learning," *Appl. Intell.*, vol. 54, no. 21, pp. 10 929–10 947, 2024.

[26] P. G. Omran, K. Wang, and Z. Wang, "Learning temporal rules from knowledge graph streams," in *Proceedings of the AAAI Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE)*. Stanford University, Palo Alto, California, USA: CEUR-WS.org, 2019, pp. 1–8.

[27] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in *International Conference on Learning Representations (ICLR)*. Virtual Event: OpenReview.net, 2021, pp. 1–24.

[28] H. Sun, J. Zhong, Y. Ma, Z. Han, and K. He, "Timetraveler: Reinforcement learning for temporal knowledge graph forecasting," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 8306–8319.

[29] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, and Y. Babaei, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, vol. abs/2307.09288, 2023.

[30] Meta Llama Team, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, vol. abs/2407.21783, 2024.

[31] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo, "Llamafactory: Unified efficient fine-tuning of 100+ language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, aug 2024, pp. 400–410.