GenTKG: Generative Forecasting on Temporal Knowledge Graph

Anonymous ACL submission

Abstract

The rapid advancements in large language models (LLMs) have ignited interest in the temporal knowledge graph (tKG) domain, where conventional embedding-based and rule-based methods dominate. The question remains open of whether pre-trained LLMs can understand structured temporal relational data and replace them as the foundation model for temporal relational forecasting. Therefore, we bring temporal knowledge forecasting into the generative setting. However, challenges occur in the huge chasms between complex temporal graph data structure and sequential natural expressions LLMs can handle, and between the enormous data sizes of tKGs and heavy computation costs of finetuning LLMs. To address these challenges, we propose a novel retrieval augmented generation framework named GenTKG combining a temporal logical rule-based retrieval strategy and few-shot parameter-efficient instruction tuning to solve the above challenges, respectively. Extensive experiments have shown that GenTKG outperforms conventional methods of temporal relational forecasting with low computation resources using extremely limited training data as few as 16 samples. GenTKG also highlights remarkable cross-domain generalizability with outperforming performance on unseen datasets without re-training, and in-domain generalizability regardless of time split in the same dataset. Our work reveals the huge potential of LLMs in the tKG domain and opens a new frontier for generative forecasting on tKGs. [1]

1 Introduction

Forecasting the future lies in the intrinsic nature of humans to take controllability over the futural uncertainty ever since the existence of ancient fortunetellers who predict the future with insights into historical events. As the wave of Artificial General Intelligence (AGI) led by Large Language Models (LLMs) (Bubeck et al., 2023) showcases a persistent craving for World Models(Matsuo et al., 2022) that can model the complex information evolving in the real world, master the implicit rules and give predictions of what might happen next based on the historical observations(Mialon et al., 2023), we term this challenge for LLMs as *Generative Forecasting*. We find Temporal Knowledge Graph (tKG) is a natural instance for investigating such a challenge attributed to the evolving world knowledge it contains and the task performed on it, namely *temporal knowledge graph forecasting*. In short sentence, tKGs are multi-relational, directed graphs with labeled timestamped edges between entities (nodes) and can be viewed as streaming data sources where events come hourly, daily, or yearly, etc., and tKG forecasting task aims to forecast future events at timestamp t based on past historical events before t. Specifically, tKG originates from Knowledge Graph (KG) (Nickel et al., 2015) which structures knowledge fact in the real world in the form of triples (e_s, r, e_o), such as *(Paris, the capital of, France)*, where e_s, e_o represent the subject and object entity respectively, and r represents the observed predicate between the two entities. As world knowledge evolves constantly over time such as the inaugurated presidents of the USA, the Temporal Knowledge Graph (tKG) was introduced by (Tresp et al., 2015) to indicate the temporal effectiveness of the world events by extending a timestamp t to form quadruples (e_s, r, e_o, t). For example, *(Donald Trump, the president of, the USA, 2021)* is followed by *(Joe Biden, the president of, the USA, 2023)*. The tKG forecasting task aims to answer queries $(e_s, r, ?, t)$ that predict the missing object given history events before t.

Conventional embedding-based graph representation learning methods (Goel et al., 2020; Han et al., 2020a; Sun et al., 2021; Yang et al., 2020) require carefully designed models that embed in-

[1] Codes and data will be released after review.

dexed quadruples into hidden latent space and hence lose the semantic aspects of events in tKGs. Besides, they require separate training for different datasets and hence suffer to handle even slight dataset modification and time split adaptation. In stark contrast, the rule-based methods (Liu et al., 2022) focus on mining temporal logic rules within the tKG graph structure in a symbolic way using neural networks. However, it posesses limited scalability to only similar datasets sharing silimar rules. With the huge advancements emerging with numerous large language models (LLMs) (Wei et al., 2022), for example utilizing the emergent in-context learning (ICL) ability of LLMs (Dong et al., 2022) by sequentializing temporal ascending ordered tKG facts to texts but failed to compete with the above conventional methods (Lee et al., 2023). The question remains open: **Can pre-trained LLMs understand structured temporal relational data and replace conventional methods as the foundation model for temporal relational forecasting?**

To address the above issue, we bring temporal knowledge forecasting into the ***generative forecasting*** setting and deliberately prioritize the most influential factors in these two domain: the temporal and structural characteristics of tKGs and the flexible natural language processing abilities of Large Language Models (LLMs). However, two challenges stand in the middle how to integrate them organically. The first is the ***modality challenge*** between data structures. As tKG are complex temporal multi-relational graph data with tens of thousands of quadruples, it is hard to adapt to sequential natural language expressions that LLMs can process. The second is the ***computation challenge*** with the enormous costs of fine-tuning LLMs especially with tens of thousands of quadruples requiring months of training time on consumable graphic cards.

To solve the above two challenges, we propose **GenTKG**, a novel *retrieval-augmented generation* framework that solves the tKG forecasting task in the *generative forecasting* setting, outperforming embedding-based, rule-based and ICL methods. Besides, GenTKG serves as an instantiation that sheds light on the promising *generative forecasting* ability of LLMs. For the first *modality challenge* between structured temporal graph data and sequential natural languages, we solve it in the retrieval phase. We utilize a temporal logical rule-based retrieval strategy (TLR) that mines the temporal logic rules of the tKGs and forms a rule bank. These rules serve to retrieve the most temporally and logically relevant historical facts to the give query. These facts are then sequentialized to natural languages in the ascending temporal order and fill in a specialized prompt template to LLMs. Although the prompts are in the form of sequential natural languages, they inherit structural information in the tKG implicitly since the extraction process are highly dependent on learned structural rules. These prompts enable LLMs to comprehend temporal relational data, and TLR enables the input window of LLM to serve as the implicit and decouplable interface for communicating temporal and structural relational data to LLM. Moreover, TLR delivers improvement over recent pure ICL method, regardless of the backbone LLM being used.

For the second *computation challenge* between huge tKG size and high computation costs of LLM, we solve it in the generation phase. We propose a few-shot parameter-efficient instruction-tuning strategy (FIT) that aligns LLM with temporal relational forecasting task and reforming it into an autoregressive generation task. We further decompose the second *computation challenge* in two subtasks from the perspective of model and data respectively. The first subtask is to deal with the enormous computation costs and hardware requirements in training LLM. We solve this subtask with a parameter-efficient fine-tuning (PEFT) adaptation method, specifically Low-rank Adaptation (LoRA)(Hu et al., 2021). The second subtask is to deal with the enormous size of training data in tKGs. We deliberately think out of the box by bypassing learning the data like conventional methods and instead, letting the LLM learn the generative forecasting task on tKG. In other words, we reform data-centric model learning to task-centric LLM alignment that aligns LLMs with tKG forecasting task through instruction tuning. We have specially designed task instruction, retrieved facts as input, and generative predictions as output. Besides, we introduce few-shot tuning that further reduces training data to only 1024 prompt-response pairs which is as few as 0.27% of original tens of thousands of training data with exceeding performance. Under extreme case, we could further reduce to as few as 16 samples which is 0.0042% of original data while maintaining comparable performance to conventional methods.

Our approach offers a foundational framework for future explorations in generative forecasting on temporal knowledge graphs. Our contributions can be summarized as follows:

1. **Opening a frontier of generative forecasting on tKG.** To the best of our knowledge, we are the first to introduce instruction-tuned generative LLM to the tKG domain. Our framework **GenTKG** proposes a novel retrieval augmented generation paradigm for tKG forecasting, regardless of the backbone LLM.

2. **Drastically low computation costs with exceeding performance.** With only 16-shots parameter-efficient instruction tuning, we can already reach comparable results to conventional methods. With 1024-shots tuning, we can outperform existing rule-based, embedding-based, and the recent in-context-learning method.

3. **Task reformulation from data learning to task alignment.** We bypass designing specific models to learn specific tKG datasets. Instead, we novelly reform the data-centric learning to task-centric LLM alignment that aligns LLMs to generative forecasting task on tKG.

4. **Generalizability across datasets without retraining**. With one-time training on a single dataset, our GenTKG has showcased remarkably both cross-domain and in-domain generalizability with exceeding performance on multiple datasets without retraining.

2 Generative Forecasting on Temporal Knowledge Graph

In this section, we explain our GenTKG framework following its two-phase methodology: Retrieve-then-Generate, in two sections. In Section 2.1, we explain the retrieval phase, which proposes a temporal logical rule-based retrieval strategy (TLR) to capture historical facts that exhibit high temporal relevance and logical coherence. In Section 2.2, we delve into the details of the few-shot parameter-efficient instruction-finetuning strategy (FIT), an essential component that aligns Large Language Models (LLMs) to the task of generative forecasting on temporal knowledge graphs.

2.1 Temporal Logic Rule-based Retrieval

The TLR retrieval strategy is inspired by the phenomenon that a pair of entities can have many interactions at different timestamps such as a president visiting the same country multiple times. Another intuition behind this is that some relations tend to be temporally and logically sequential, for example in ICEWS14 we can see *(Angela Merkel, discuss by telephone, Barack Obama, 2014/07/22)* and *(Angela Merkel, consult, Barack Obama, 2014/08/09)*. Therefore, we borrow a partial idea of TLogic(Liu et al., 2022) that mines the temporal logic rules hidden in the tKG structure. Notably, we opt to choose rules with a length equal to one that complies with the input context constraints of the LLMs, and don't apply rules directly for ranking each entity. Then we propose the TLR that retrieves the most temporally related and logically supportive history events for the given query based on these learned rules. To help understand our retrieval strategy, two definitions and the algorithm are given in the following.

Definition I (Temporal Random Walk) A non-increasing temporal random walk W starting from subject entity $e_s \in \mathcal{E}$ to object entity $e_o \in \mathcal{E}$ in the tKG $\mathcal{G}$ is defined as a cycle of edges $((e_s, r_1, e_o, t_2), (e_s, r_2, e_o, t_1))$ with $t_2 > t_1$ where $(e_s, r_i, e_o, t_i) \in \mathcal{G}$ and $i \in 1, 2$. The time constraints ensure that the edges are traversed only backward in time.

Definition II (Temporal Logical Rule) A cyclic temporal logical rule R is defined as $(E_1, r_h, E_2, T_2) \leftarrow (E_1, r_b, E_2, T_1)$ with $T_2 > T_1$, where E_i and T_i for $i \in 1, 2$ are replaceable variables that represent entities and timestamps. The left-hand side of R is called the rule head, with r_h being the head relation, while the right-hand side is called the rule body, with r_b being the body relation. A rule head can be supported by multiple rule bodies denoting different rules as $\mathcal{TR}$. A $\mathcal{TR}$ implies that if the rule body holds then the rule head is true for a future timestamp T_2. The confidence of a rule conf($\mathcal{TR}$) is defined as dividing the rule support by the body support, where the support is the number of quadruples satisfying rule bodies or rule heads with time constraints within $\mathcal{TR}$.

Rule Learning Let r_h be a fixed relation, for which we want to learn rules. We sample an edge (e_1, r_h, e_2, t), which will serve as the rule head, uniformly from all edges with relation r_h. Then the temporal random walker samples iteratively candidate edges adjacent to the current object $\mathcal{C}(e_2, t) := \{(e_2, r, e_1, \hat{t}) \mid (e_2, r, e_1, \hat{t}) \in \mathcal{G}, \hat{t} < t\}$, where $\hat{t}$ is the timestamp associated with the next transition

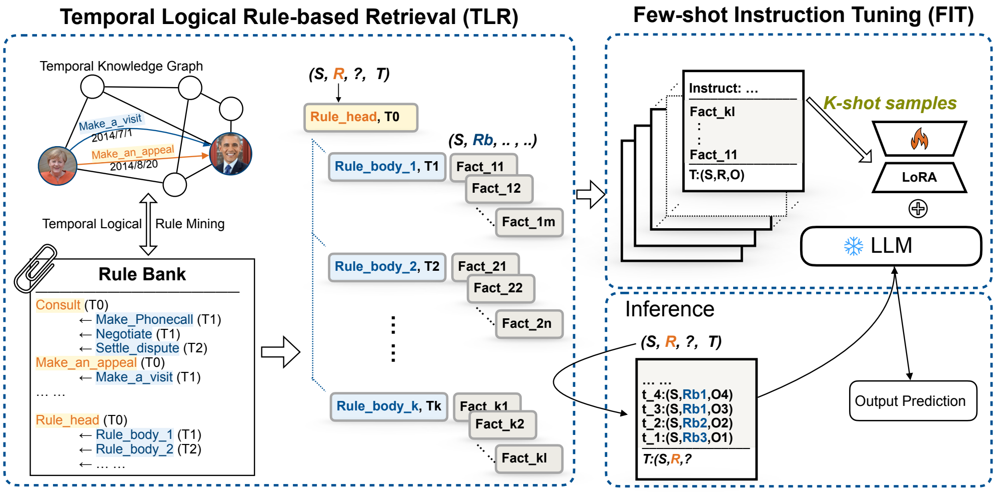


Figure 1: Framework of GenTKG. GenTKG first retrieves relevant facts based on a temporal logical rule-based retrieval strategy, then samples K prompts for few-shot parameter-efficient instruction-tuning of LLM that aligns LLM to the task of generative temporal knowledge graph forecasting.

edge. Besides, we use an exponentially weighted transition distribution that prioritizes temporally closer edges during sampling which is defined as

$$\mathbb{P}\left(u; e_2, t\right) = \frac{\exp\left(t_u - t\right)}{\sum_{\hat{u}\in\mathcal{C}(e_2,t)} \exp\left(t_{\hat{u}} - t\right)} \quad (1)$$

where t_u denotes the timestamp of edge u. After a fixed sampling we can collect a set of temporal walks satisfying the rule $(E_1, r_h, E_2, T_2) \leftarrow (E_1, r_b, E_2, T_1)$. We then estimate the confidence of the rules following the definition II.

Temporal Logic Rule-based Retrieval After gaining learned temporal logical rule sets, we order them according to the associated confidence scores. For a given forecast query $(e_s, r, ?, t)$ we retrieve a candidate subgraph $\mathcal{G}_s(e_s, r, t)$ from the TKG $\mathcal{G}$ containing temporally and logically relevant histories for the given query, with respect to the subject entity, relation, and timestamp. Since the query subject entity is fixed, there are two key factors in the retrieval algorithm, i.e. time window and rule grounding. First, we define the time window as $TW = [t_-, t]$ with $t_- := t - w$, where the $w \in \mathbb{N}^+$ represents the time window length backward starting from the query timestamp. The maximum length of w is $\min\{t_{max}, t\}$ with t_{max} denoting the maximum timestamp of the datasets. Second, the query relation r is fixed as a rule head r_h. Within each TW, we apply the learned rules $\mathcal{TR}$ and select top k various rule bodies $r_{b_1}, r_{b_2}, \cdots, r_{b_k}$ regarding to r in descending confidence and add historical events $(e_s, r_b, e_o, t - w)$ to $\mathcal{G}_s(e_s, r, t)$ for the given query. The size of $\mathcal{G}_s(e_s, r, t)$ can be adjusted dynamically with respect to w and k. We stop the retrieval until a maximum history length N is reached. For instance, we retrieve history events iteratively with the descending confident rule bodies for each time window backtrace step until a maximum history length of 50 is reached. At the end of the retrieval phase, we reorder all history events in temporal descending order for each query.

2.2 Align LLM to Generative tKG Forecasting

In the second phase of the proposed GenTKG framework, we contribute to transforming the conventional data-centric tKG model learning task into an alignment task that aligns LLM with generative forecasting on tKGs. We utilize a few-shot parameter-efficient instruction tuning strategy (FIT) under the settings of low GPU resource consumption with a single graphic card. In 2.2.1, we first describe the instruction prompt design. In 2.2.2, we describe the parameter-efficient instruction tuning for training our generative model. In 2.2.3 , we explain the few-shot tuning strategy that efficiently aligns the LLM with temporal relational forecasting with as few as 1024 samples and explore the lower-bound of samples for few-shot tun-

PART	CONTENT
Task Instruction	You must be able to correctly predict the next {object_label} from a given text consisting of multiple quadruplets in the form of "{time}:[{subject}, {relation}, {object_label}.{object}]" and the query in the form of "{time}:[{subject}, {relation}," in the end. You must generate only the single number for {object_label} without any explanation.
Task Input	93: [Abdulrahman, Make_statement, 8092.Government_(Nigeria)] 113: [Abdulrahman, Make_statement, 8092.Government_(Nigeria)] 162: [Abdulrahman, Praise_or_endorse, 15546.Muslim_(Nigeria)] 197: [Abdulrahman, Consult, 8488.Governor_(Nigeria)] 197: [Abdulrahman, Make_statement, 8092.Government_(Nigeria)] 228: [Abdulrahman, Praise_or_endorse, 15414.Muhammadu_Buhari] 270: [Abdulrahman, Make_an_appeal_or_request, 3835.Citizen_(Nigeria)] 270: [Abdulrahman, Praise_or_endorse,
Task Output	3835.Citizen_(Nigeria)]

Figure 2: Instruction Prompt Design

ing. In 2.2.4, we describe the inductive generalizatoin ability of generative forecasting on tKG.

2.2.1 Instruction Prompt Design

Instruction Tuning is a crucial technique that finetunes LLMs with human-curated instruction and response pairs as the training data, empowering LLMs with instruction-following capability (Zhou et al., 2023). The construction of an instruction sample is usually composed of three parts, i.e. task instruction, task input, and task output. Task instruction clarifies the definition of the task for LLMs to comprehend and gives explicit solutions for LLMs to follow and execute. Task input in natural languages is input data serving as context for LLMs. Task output is the decoding results based on the input prompt. In our proposed GenTKG framework, we adapt the temporal knowledge graph forecasting task to the instruction task for LLMs with individual adaptation for the three parts partially following the setting in (Lee et al., 2023). As a demonstration, the instruction is depicted in Figure 2. Except for the designed task instruction, the task input is modeled as ordered historical events retrieved from the TLR phase for a given query (e_s, r, e_o, t) as described in 2.1. Each fact is filled in the template of "$t : [e_s, r, n_{e_o}.e_o]$". The query (e_s, r, e_o, t) is expressed in a similar but partial way as "$t : [e_s, r,$" for LLM to complete as generative predictions. It is worth noting that we conserve the format in (Lee et al., 2023) that maps each candidate object e_o with a numerical index n_{e_o} as a fair comparison. However, (Lee et al., 2023) try to avoid unfair tokenization for different entities with this index and use the probabilities of index tokens generated by the LLMs to get ranked scores of output entities in an indirect way. But this can only been used on GPT-like model and cannot handle LLaMA-like models harnessing individual tokenization. Therefore we use top generated entity names directly for prediction evaluation.

2.2.2 Parameter-efficient Instruction Tuning

Direct fine-tuning of the entire model is computationally demanding and time-consuming. To address these computational challenges, we adopt the Low-Rank Adaptation (LoRA) technique (Hu et al., 2021). LoRA involves the freezing of pre-trained model parameters $\boldsymbol{\theta_0}$ while introducing trainable additional parameters $\boldsymbol{\theta_0}$ that can be decomposed into low-rank matrices $\Delta\boldsymbol{\theta}_0 = \mathbf{BA}, \mathbf{B} \in \mathbb{R}^{d\times r}, \mathbf{A} \in \mathbb{R}^{r\times k}, r \ll \min(d,k)$ that incorporat supplimentary information to the LLM.

At present, there are large amounts of LLMs released, such as GPT series (Kojima et al., 2022; Radford et al., 2019), T5 series (Raffel et al., 2020), CHinchilla (Hoffmann et al., 2022), and LLaMA (Touvron et al., 2023), etc.. Among these, proprietary models can only be accessed by APIs such as ChatGPT with limited adaptation and alignment possibilities that hinder the research purpose. To facilitate the research of generative forecasting on temporal knowledge graph, we carefully opt for the open-sourcing LLMs, i.e. GPT-NeoX-20B (Black et al., 2022) and LLaMA2-7B(Touvron et al., 2023), which is the third-party reproduction of GPT-3 and open-source public model respectively. Due to hardware limitations, we leave GPT-NeoX-20B frozen to investigate the effectiveness of our retrieval phase through its in-context learning ability. We perform the whole GenTKG framework on LLaMA2-7B with consumable adaptation.

2.2.3 Efficient Alignment with Few-shot Tuning

Our framework contributes a remarkably efficient and effective few-shot training strategy. The hypothesis has been proven that alignment can be a simple process where the LLMs learn the style or format for responding to prompts and expose the knowledge and capabilities that were already acquired during pretraining (Zhou et al., 2023). Therefore, considering the volume of temporal knowledge graphs that usually possess tens of thousands of training data, we propose a K-shot tuning paradigm where only an extremely limited number of K samples are uniformly sampled from the temporal-ordered training set for language model adaptations. In our case, we select only 1024 samples which takes up as few as 0.27% of the original GDELT dataset sizes that conventional methods usually fully trained on. We further prove that our method can acquire temporal relational forecasting capability rapidly with severely limited training data (0.0027%) with an extreme 16-shot training setting while maintaining comparable performances to conventional method.

2.2.4 Inductive Setting

Due to the novel transformation from data-centric learning to task-centric alignment which forces the LLM is aligned to the temporal relational forecasting task itself rather than the learning of the tKG data. GenTKG also delivers remarkable generalizaibility in various inductive settings.

Cross-domain generalizability. LLM trained on one dataset can be inferred directly on other datasets. An inductive GenTKG only requires learning the temporal-logical rule-based retrieval strategy for the new datasets in the first phase to ensure proper prompts with relevant histories. However, it doesn't require retraining LLM in the second phase. Still, high-performance gains are maintained and even comparable to the original setting.

In-domain generalizability. GenTKG maintains high-performance gains on the same dataset even trained on only partial training data. The partition can be limited to a small fraction such as 5% of original training data. This characteristic exceeds conventional methods which always suffer drastic performance drops even with minor change of critical value of the forecasting timestamp between the train and evaluation set.

3 Experimental Setup

In this section, we describe the experimental setup of GenTKG framework. Specifically, we describe four datasets, the evaluation protocols, and the experimental design.

Datasets Four benchmark datasets are used to evaluate GenTKG: 1) ICEWS14 (Boschee et al., 2015) 2) ICEWS18 (Boschee et al., 2015) 3) GDELT (Leetaru and Schrodt, 2013) 4) YAGO (Mahdisoltani et al., 2013). The two versions of the Integrated Crisis Early Warning System (ICEWS) both consist of timestamped political events, e.g., (Angela Merkel, visit, India, 2015-03-25). The GDELT and YAGO datasets are extracted from the subsets of GDELT and YAGO knowledge bases containing facts and time information. Dataset statistics is shown in Table 4 in the Appendix.

Evaluation Since GenTKG generates entity predictions directly, we use the temporal-aware filtred Hits@1/3/10 metric to evaluate the model performance on extrapolated link prediction. Hits@1/3/10 denotes the proportion of the actual missing entities ranked within the top 1/3/10.

Baselines Since GenTKG is the first method to introduce instruction-tuned generative models into the tKG forecasting domain, it is necessary to include three typical types of existing methods as baselines. The first are embedding-based methods, represented by RE-GCN (Li et al., 2021), xERTE (Han et al., 2020a), TANGO (Han et al., 2021), and Timetraveler (Sun et al., 2021). The rule-based method is TLogic (Liu et al., 2022) and the third type is the LLM-based ICL method with frozen parameters (Lee et al., 2023).

Experiment Design In order to comprehensively analyze GenTKG compared to different conventional methods, there are three research questions to be answered. RQ1: How is the overall performance of the proposed GenTKG framework compared with the existing conventional embedding-based, rule-based TKG methods and LLM-based ICL method? RQ2: How well is the cross-domain and in-domain generalizability of GenTKG on different inductive settings? RQ3: How do the components of the GenTKG affect its effectiveness?

4 Experimental Results

4.1 Main Results

Experiment results can be seen on Table 1. To answer the first question RQ1, our results achieve state-of-the-art performance, surpassing all three types of existing conventional including embedding-based models, rule-based method, and LLM-based in-context learning method across four datasets regarding metric Hit@1 and Hit@3 while maintaining comparable results regarding Hits@10. Our method demonstrates the promising trend for retrieval-augmented LLMs to serve as the foundation model for temporal relational forecasting, opening a new frontier in the TKG domain. More detailed results and analyses are presented in the following. We refer to GenTKG utilizing LLaMA2-7B as instantiation unless otherwise specified.

Compared to embedding-based models. For all datasets, GenTKG outperforms its best embedding-based model xERTE on ICEWS14, ICEWS18, GDELT, and Timetraveler on YAGO. Specifically, the highest performance gain is observed on GDELT with more than 58% higher on Hits@1. It is natural to conclude that GenTKG can outperform embedding-based methods.

Compared to the rule-based model. Compared

Table 1: Temporal link prediction results: Hits@1/3/10(%). The best results among each metric except for the inductive setting are highlighted in bold and the second bests are underlined.

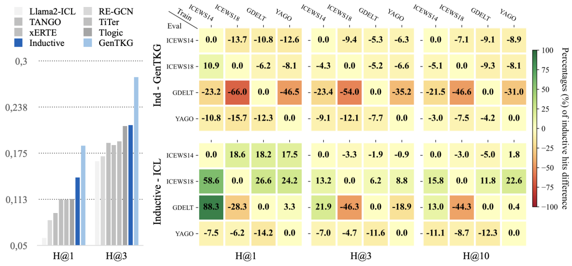
Method Type	Datasets / Models	ICEWS14 Hits@1	ICEWS14 Hits@3	ICEWS14 Hits@10	ICEWS18 Hits@1	ICEWS18 Hits@3	ICEWS18 Hits@10	GDELT Hits@1	GDELT Hits@3	GDELT Hits@10	YAGO Hits@1	YAGO Hits@3	YAGO Hits@10
Embedding-based	RE-GCN	31.3	47.3	**62.6**	22.3	**36.7**	**52.5**	8.4	17.1	29.9	46.8	60.7	72.9
	xERTE	33.0	45.4	57.0	20.9	33.5	46.2	8.5	15.9	26.5	56.1	72.6	78.9
	TANGO	27.2	40.8	55.0	19.1	31.8	46.2	9.4	18.9	32.2	56.6	65.1	71.8
	Timetraveler	31.9	45.4	57.5	21.2	32.5	43.9	11.2	18.6	28.5	60.4	77.0	83.1
Rule-based	TLogic	33.2	47.6	60.2	20.4	33.6	48.0	11.3	21.2	**35.1**	63.8	65.0	66.0
In-Context Learning	GPT-NeoX-20B	32.6	44.0	54.2	18.2	29.5	41.4	6.8	12.0	21.1	72.6	81.0	84.6
	Llama2-7B	25.8	43.0	51.0	13.5	27.6	32.6	3.6	12.5	22.0	67.7	79.0	81.8
GenTKG	GPT-NeoX-20B + TLR	35.0	47.4	57.5	21.1	33.9	45.6	10.2	16.7	27.3	73.6	**83.0**	**86.8**
	Llama2-7B + GenTKG	**36.85 ± 0.75**	**47.95 ± 0.75**	53.5 ± 0.8	**24.25 ± 0.75**	36.25 ± 1.25	42.1 ± 1.1	**13.9 ± 0.5**	**22.55 ± 0.55**	30.45 ± 0.45	**79.15 ± 2.25**	**83.0 ± 1.7**	84.25 ± 1.55
	Llama2-7B - inductive	-	-	-	22.75 ± 0.65	36.2 ± 0.7	44.0 ± 0.8	13.75 ± 0.95	20.35 ± 1.05	27.6 ± 0.8	68.9 ± 0.6	75.45 ± 0.35	82.05 ± 0.35

to the rule-based model TLogic, GenTKG outperforms TLogic on Hits@1 and Hits@3 while maintaining comparable performance regarding Hits@10 on GDELT. The slight drops in Hits@10 on ICEWS14 and ICEWS18 are because TLogic is carefully designed on these datasets while our method has more generalizability and demonstrated better performance regarding accuracy than recall.

Compared to in-context-learning method. We analyze the performance of GenTKG on different Language Model instantiations, i.e. GPT-NeoX-20B and LLaMA2-7B respectively. For GPT-NeoX-20B, we apply only the first retrieval phase of GenTKG due to hardware limitations. However, an average 10% performance increase is observed for all three metrics on all datasets even with pure retrieval-augmented in-context learning. For LLaMA2-7B, the performance gain of Hits@1 has increased remarkably even outperforming GPT-NeoX-20B which has two times more parameters, indicating the potential for greater performance of our proposed GenTKG framework if applied to larger language models.

4.2 Cross-domain Generalization

To answer the second question of GenTKG's performance in the inductive setting, the empirical results indicate that the GenTKG framework manifests a substantial capability for cross-dataset generalization. Specifically, once the LLM has been aligned to the tKG forecasting task in the second phase on any dataset, the LLM can be applied directly to any other datasets. Therefore, on a new dataset, GenTKG only requires dataset-specific temporal-logical rule-based retrieval to formulate proper prompts from the first phase, and can directly infer the predictions without retraining in the second phase. As shown in Figure 3(a), all methods are



(a) Single Dataset Evaluation. (b) Cross-checking Evaluation.

Figure 3: Cross-Domain Inductive Setting. (a) Single dataset evaluation. All training and evaluation is on GDELT except inductive GenTKG is trained on ICEWS14. (b) Cross-checking. We cross-check the trained LLaMA2 in GenTKG on different training datasets and evaluation datasets. The performance drop compared to the original training setting takes up only small percentages. Even higher performance than ICL can be observed. Absolute difference value is given on Appendix 2, explaining the huge relative difference on GDELT is due to its poor baseline performances.

trained and evaluated on GDELT, except that the LLM in inductive GenTKG is trained ICEWS14. Still, the inductive GenTKG delivers comparable performance metrics on GDELT to conventional methods with a minor performance drop compared to the orginal trained GenTKG. We further demonstrate similar inductive results by cross-checking the training and evaluation datasets as shown in Figure 3(b). Although the LLM is trained exclusively on one dataset, it still delivers comparable metrics on disparate datasets, closely approximating the outcomes of methods that were trained specifically on the identical evaluation dataset. This notable characteristic implies that the GenTKG framework is effectively capturing the underlying task-related features, as opposed to merely carefully-designed for the dataset data, a limitation commonly shared in conventional methods.

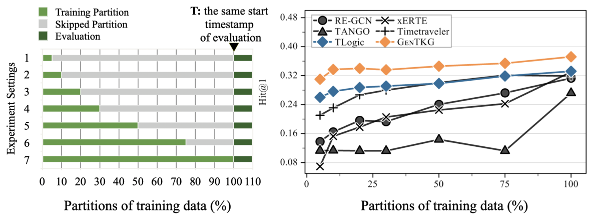


Figure 4: In-domain generalizability. GenTKG exceeds conventional methods on all different partitions of training data on ICEWS14. Values in Appendix Table 3.

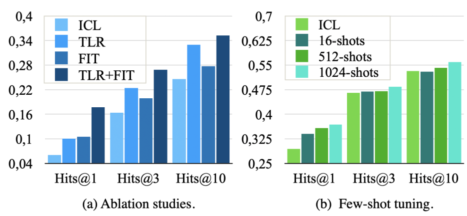


Figure 5: (a) Both TLR and FIT phases contribute to GenTKG. (b) Increasing the few-shot training parameter K improves performance.

4.3 In-domain Generalization

Apart from cross-domain generalizability, how well does GenTKG generalize to different training partitions within the same dataset? To investigate such a problem, we carefully designed various partitions of time-ordered training data ranging in {5%, 10%, 20%, 30%, 50%, 75%, 100%}. All models trained on different training partitions are evaluated on the same evaluation set starting from the same timestamp. According to Figure 4, experiments have shown that conventional methods suffer from insufficient training data while GenTKG remains exceeding performance even with as few as 5% training data. This further proves that GenTKG successfully transforms conventional data-centric learning to the task-centric alignment of LLMs and overcomes the prediction instability under the changing value of time split in forecasting setting.

4.4 Ablation study

We undertake the ablation studies on ICEWS14 to evaluate the contribution of each phase in GenTKG with three distinct variants of the GenTKG: TLR, FIT, and TLR+FIT configurations. Here, TLR represents the variant that exclusively employs temporal logical rule-based retrieval on top of ICL learning, FIT denotes the variant solely implementing few-shot parameter-efficient instruction tuning with naive fact retrieval (Lee et al., 2023), and TLR+FIT encapsulates the integration of all components within GenTKG. Figure 5(a) draws the conclusion that both phases in GenTKG framework contribute to distinct performance improvements. The whole pipeline enables GenTKG the ability to outperform existing methods.

4.5 Few-shot Tuning

To delve further into the impact of sample size within the few-shot tuning, we conducted a series of experiments on the ICEWS14 dataset employing a range of shot sizes K from the set $\{6, 512, 1024\}$. For each configuration, we employed uniform sampling on the temporally-ordered training dataset. Empirical results indicate a consistent trend of performance improvement correlating proportional to the increase in the number of training samples, as visualized in Figure 5(b). Remarkably, our findings suggest that the GenTKG framework is capable of outperforming naive ICL method even when as few as 16 shots are used for tuning. This notable finding unlocks significant potential for GenTKG in the context of aligning LLMs with temporal relational forecasting tasks from the perspective of efficient alignment or a larger scale.

5 Conclusion

In this paper, we raise the question and prove that pre-trained LLMs can understand structured temporal relational data and replace existing tKG models as the foundation model for temporal relational forecasting task. We propose a retrieval-augmented generative framework GenTKG that can efficiently align LLM with temporal relational task through two stages: temporal logical rule-based retrieval and few-shot parameter-efficient fine-tuning. Extensive experimental results demonstrate that GenTKG framework outperforms conventional embedding-based, rule-based and ICL methods. Moreover, GenTKG is training-light through consumable computation resources with extremely few training data, and exhibits strong cross-domain and in-domain transferability breaking the barriers of conventional data-centric learning.

6 Limitations

GenTKG is limited by the input context window of LLMs. Specifically, for LLaMA2, the input context window is 4096 tokens with an average upper length limit of 50 history facts that limit the performance of Hit@10. We leave this to future work.

Ethics Statement

GenTKG is tailored to generative forecasting on temporal knowledge graph and can be applied to a wide variety of downstream tasks with generative forecasting setting, such as recommendation system, anomly detection, etc. It can also power search and serve to improve users' lives. GenTKG can help protect data with its generalizability which requires less training over various datasets. The risk of GenTKG might comes from risks inherited in open-source LLMs, such as hallucinations.

Liscence

The datasets used in this research work is open-sourced and can be seen on references. We derive some datasets from the original version within the intended use term. The code and source will be released after review.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, Toronto, Canada. Association for Computational Linguistics.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Zifeng Ding, Jingpei Wu, Bailan He, Yunpu Ma, Zhen Han, and Volker Tresp. 2022. Few-shot inductive learning on temporal knowledge graphs using concept-aware information. *arXiv preprint arXiv:2211.08169*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020a. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8352–8364.

Zhen Han, Ruotong Liao, Beiyan Liu, Yao Zhang, Zifeng Ding, Jindong Gu, Heinz Koeppl, Hinrich Schuetze, and Volker Tresp. 2022. Enhanced temporal knowledge embeddings with contextualized language representations.

Zhen Han, Yunpu Ma, Peng Chen, and Volker Tresp. 2020b. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. *arXiv preprint arXiv:2011.03984*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

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

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.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4120–4127.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. 2022. Deep learning, reinforcement learning, and world models. *Neural Networks*, 152:267–275.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Md Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. 2022. Dialokg: Knowledge-structure aware task-oriented dialogue generation. *arXiv preprint arXiv:2204.09149*.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. *arXiv preprint arXiv:2109.04101*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, and Denis Krompaß. 2015. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jinzhu Yang, Wei Zhou, Lingwei Wei, Junyu Lin, Jizhong Han, and Songlin Hu. 2020. Re-gcn: relation enhanced graph convolutional network for entity alignment in heterogeneous knowledge graphs. In *Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part II 25*, pages 432–447. Springer.

Miao Zhang, Rufeng Dai, Ming Dong, and Tingting He. 2022. Drlk: dynamic hierarchical reasoning with language model and knowledge graph for question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5123–5133.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A Related Works

Temporal Knowledge Graphs Temporal knowledge graphs (tKGs) are multi-relational, directed graphs with labeled timestamped edges between entities (nodes). Let $\mathcal{E}$ and $\mathcal{P}$ represent a finite set of entities and predicates. A quadruple (e_s, r, e_o, t) represents a timestamped and labeled edge between a subject entity $e_s \in \mathcal{E}$ and an object entity $e_o \in \mathcal{E}$ at a timestamp $t \in \mathcal{T}$. Let $\mathcal{F}$ represent the set of all true quadruples, i.e., real events in the world, the temporal knowledge graph forecasting is the task of predicting missing object entity at timestamp t, i.e. $(e_s, r, ?, t)$ based on a set of observed facts $\mathcal{O}$ before t, which is a subset of $\mathcal{F}$. Current methods can be categorized into two streams. On the one hand, embedding-based models learn representations of the quadruples with carefully designed embedding models(Han et al., 2020a; Goel et al., 2020; Sun et al., 2021; Han et al., 2020b; Ding et al., 2022). On the other hand, the rule-based methods mine the temporal logical rules extracted and extract candidates directly on the temporal knowledge graphs(Liu et al., 2022).

Investigating TKG with Language Models The semantic part stored in the temporal knowledge graphs is heavily overlooked in either embedding-based or rule-based temporal knowledge graph methods. Early explorers had tryouts in introducing language models in the TKG domain, some fused pre-trained language representations to the temporal knowledge embeddings (Han et al., 2022), and some flattened explicit temporal events with the emergent in-context learning ability of large language models however not comparable with conventional performance (Lee et al., 2023). Other researchers had tryouts in combing KG with LLM, utilizing the knowledge-aware prompting method (Baek et al., 2023; Rony et al., 2022; Sun et al., 2023; Zhang et al., 2022), however, cannot be transferred to the tKG domain due to their ignorance of temporal characteristics.

B Supplimentary Materials

B.1 Implementation details.

Experiment hyperparmeters will be release in code after review. We run experiments 3 times and take averages with A40 GPU.

Table 2: Appendix table for inductive result differences of GenTKG compared with non-inductive GenTKG (-nind) results and compared with ICL on LLaMA2 with baseline retrieval (-ICL).

	Train / Eval	Hits@1 ICEWS14	ICEWS18	GDELT	YAGO	Hits@3 ICEWS14	ICEWS18	GDELT	YAGO	Hits@10 ICEWS14	ICEWS18	GDELT	YAGO
$\Delta(-nind)$	ICEWS14	-	-0.05	-0.04	-0.05	-	-0.05	-0.03	-0.03	-	-0.04	-0.05	-0.05
	ICEWS18	0.02	-	-0.02	-0.02	-0.02	-	-0.02	-0.02	-0.02	-	-0.04	-0.04
	GDELT	-0.04	-0.12	-	-0.09	-0.07	-0.15	-	-0.10	-0.08	-0.17	-	-0.11
	YAGO	-0.08	-0.11	-0.09	-	-0.07	-0.09	-0.06	-	-0.02	-0.06	-0.03	-
$\Delta(-ICL)$	ICEWS14	-	0.05	0.05	0.04	-	-0.01	-0.01	0.00	-	-0.02	-0.03	0.01
	ICEWS18	0.08	-	0.03	0.03	0.04	-	0.02	0.02	0.05	-	0.04	0.07
	GDELT	0.05	-0.02	-	0.00	0.04	-0.08	-	-0.03	0.03	-0.11	-	0.00
	YAGO	-0.05	-0.04	-0.09	-	-0.05	-0.04	-0.09	-	-0.09	-0.07	-0.10	-

Table 3: Appendix table for few-shot results of conventional methods and GenTKG.

	Top 5% Hits@1	Hits@3	Hits@10	Top 10% Hits@1	Hits@3	Hits@10	Top 20% Hits@1	Hits@3	Hits@10	Top 30% Hits@1	Hits@3	Hits@10	Top 50% Hits@1	Hits@3	Hits@10	Top 75% Hits@1	Hits@3	Hits@10	Top 100% Hits@1	Hits@3	Hits@10
RE-GCN	13.79	22.09	30.27	16.47	25.23	34.19	19.63	29.67	39.83	19.30	30.66	42.97	24.05	36.72	48.84	27.23	40.42	54.04	31.30	47.30	62.60
xERTE	06.95	14.17	25.46	15.27	26.79	39.43	17.80	29.26	42.08	20.56	31.39	43.63	22.51	34.15	46.59	24.25	36.07	48.27	33.00	45.40	57.00
TANGO	11.29	17.18	22.97	11.34	17.47	22.98	11.25	17.38	23.38	11.25	17.39	23.40	14.37	17.51	22.77	11.25	16.90	22.50	27.20	40.80	55.00
Timetraveler	21.06	34.78	49.10	23.10	35.71	49.96	26.69	39.42	51.78	27.98	40.14	53.23	30.05	42.82	54.74	32.11	45.33	57.14	31.90	45.40	57.50
TLogic Original	26.03	37.42	46.50	27.65	39.55	48.72	28.72	40.48	50.71	29.11	41.79	51.90	29.84	42.40	53.37	31.89	45.01	57.37	33.20	47.60	60.20
GenTKG	30.60	42.20	49.30	34.00	45.40	52.10	34.90	46.60	54.00	34.70	46.90	54.40	36.00	48.70	55.50	36.50	48.30	55.30	37.20	48.80	56.30

Table 4: Dataset statistics.

Datasets	#train	#valid	#test	#entity	#relations	time gap
ICEWS14	74854	8514	7371	7128	230	1 day
ICEWS18	373018	45995	49545	23033	256	1 day
GDELT	79319	9957	9715	5850	238	15 mins
YAGO	220393	28948	22765	10778	23	1 year