SALMON: A Structure-Aware Language Model with logicality and densification strategy for Temporal Knowledge Graph Reasoning

Fu Zhang[†], **Jinghao Lin**[†], **Jingwei Cheng**[*]
School of Computer Science and Engineering, Northeastern University, China
`{zhangfu,chengjingwei}@mail.neu.edu.cn; linjinghao2k@gmail.com`

Abstract

Temporal knowledge graph reasoning (TKGR) is a crucial task that involves reasoning at known timestamps to complete the future facts and has attracted more and more attention in recent years. The current TKGR models are mainly based on graph neural networks or tensor decomposition techniques. Few works in TKGR focus on pre-trained language models (PLMs) which have powerful sequence modeling capabilities to capture the temporal associations between facts. In this paper, we propose a model SALMON: a **S**tructure-**A**ware **L**anguage **M**odel with l**o**gicality and de**n**sification strategy. Specifically, we design a PLM-based framework with a structure-aware layer inside to jointly capture the temporal evolving pattern and structural information in TKGs. To further enhance the model's ability to infer causal associations of facts, we propose a logical judging module, which can guide the model to prioritize learning the most relevant evolving information of logical causal associations in TKGs during the training process. Moreover, we propose a densification strategy based on large language models, through a carefully crafted Chain of Thought prompt, to dig out some knowledge necessary for reasoning about fact associations, thereby making the model perform better. Extensive experimental results demonstrate the superiority of our model over the state-of-the-art baselines.[1]

1 Introduction

Knowledge graphs (KGs) (Hogan et al., 2021) represent relations among real-world entities, enabling a broad spectrum of applications in natural language processing tasks. As knowledge evolves over time, timestamps are incorporated into KGs, giving rise to temporal KGs (TKGs) (Cai et al., 2023), which play a pivotal role in supporting applications that necessitate a temporal dimension in their knowledge representation. Formally, a TKG is a graph-structured dataset encompassing numerous time-sensitive relational facts. These facts (also called *events*) are expressed as quadruples (subject entity, relation, object entity, timestamp), denoted as $(s,\ r,\ o,\ t)$. For example, the event (Trump, Sign_formal_agreement, Canada, 2018-10-17), together with the events causally associated with it as shown in Figure 1, embody the temporal aspect of knowledge. With the widespread application of TKGs, *temporal knowledge graph reasoning* (TKGR) (Cai et al., 2023) is proposed as a crucial task that involves reasoning at known timestamps to complete future facts.

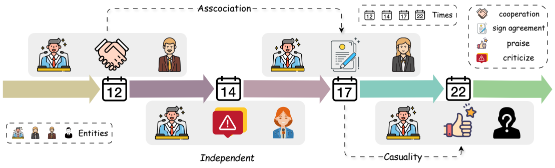


Figure 1: A chronologically ordered timeline of events extracted from a TKG ICEWS18 (García-Durán et al., 2018) illustrates the actions of President Trump.

Many previous efforts have been devoted to representation learning for KGs. In recent years, some methods based on *temporal representation learning* have been introduced for TKGR. Examples of these methods include tensor decomposition-based TNTComplEx (Lacroix et al., 2020) and graph neural network-based TeMP (Wu et al., 2020) (see Section 2 in details). The fundamental idea is to project entities and relations in a TKG into a low-dimensional vector space, preserving the structure and temporal patterns of the TKG.

Moreover, in order to further grasp the causal associations among events, another type of approaches based on *rule* such as XERTE (Han et al., 2020), Tlogic (Liu et al., 2022) have also been

---

[1] Code is available at `https://github.com/linjh1118/SALMON`.
[†] Equal contribution. [*] Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 8761–8774
November 12-16, 2024 ©2024 Association for Computational Linguistics

proposed. These methods mainly walk on TKGs through rules (the walking paths constitute a reasoning chain), and then reach the final entity to achieve TKGR. Although the rule-based method can provide explainable reasoning, this walking method has relatively high requirements on the integrity of the reasoning chain in TKGs. When a key clue event related to the entity to be predicted is missing, the reasoning may be interrupted.

In addition, with the excellent performance of language models (e.g., BERT (Kenton and Toutanova, 2019), GPT (Radford et al., 2018)) in natural language processing tasks, TKGR based on *language models* has gradually attracted attention. The earlier work KG-BERT (Yao et al., 2019) and LASS (Shen et al., 2022) attempt to leverage pre-trained language models (PLMs) to address traditional KG reasoning task by treating triples as textual sequences. The recent work PPT (Xu et al., 2023) utilizes a PLM with prompts to integrate temporal KG information into language models for achieving TKGR. Due to the sparsity of TKGs, many of the facts are relatively independent and lack historical information related to them as shown in Figure 1. This limits the ability of PPT to fully explore the deeper reasons for associations among facts to a certain extent, resulting in limited performance. The potential of language models in TKGR task still has a great space to explore.

To address these issues, in this paper we propose a novel model, **SALMON** (**S**tructure-**A**ware **L**anguage **M**odel with **LO**gicality and De**N**sification Strategy), which leverages the strengths of PLMs to enhance TKGR. Built on top of the PLM-based framework, SALMON includes a dedicated *structure-aware layer* to jointly capture the evolving patterns and structural information in a TKG, leading to a more comprehensive understanding of the TKG. Further, to enhance the model's reasoning capabilities, we design a *logical judging module*. Instead of traditional rules-based walking, our logical judgment module aims to give priority to learning the most relevant evolving information of logical causal associations in the TKG during the training process of PLMs. To alleviate the sparsity of TKGs mentioned above, inspired by the potential of recent large language models (LLMs), we propose a *densification strategy* that harnesses the power of LLMs through a carefully designed prompt, to enhance the TKGR performance by digging out some evidence necessary for reasoning about associations among events. Our contributions can be summarized as follows:

- We propose a PLM-based framework tailored for TKGR, incorporating a well-designed structure-aware layer to capture both temporal evolution and structure characteristics of TKGs.
- We design a logical judging module, which guides the model to prioritize the most relevant evolving information of logical causal associations in TKGs for reasoning.
- We propose a LLMs-based densification strategy that harnesses the power of LLMs through a carefully designed prompt, aims to mine some evidence necessary to reason about events with little historical information.
- To validate the efficacy of SALMON, we conduct extensive experiments on three datasets, and the results show that our SALMON achieves state-of-the-art performance for temporal knowledge graph reasoning tasks.

2 Related Work

2.1 Representation learning-based models

Previous research has predominantly centered around the methodologies for capturing structural and temporal information to facilitate inference. In some instances, TA-DistMult (García-Durán et al., 2018), TNTComplEx (Lacroix et al., 2020), TIMEPLEX (Jain et al., 2020) and TeLM (Xu et al., 2021) integrate conventional tensor decomposition techniques (e.g., DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016)) with the incorporation of temporal information. Additionally, TeMP (Wu et al., 2020), CyGNet (Zhu et al., 2021), GTRL (Tang and Chen, 2023) and SiMFy (Liu et al., 2023) based on graph neural networks, explore the segmentation of quadruples within TKGs based on temporal blocks.

2.2 Rule-based models

An alternative paradigm, exemplified by rule-based approaches such as XERTE (Han et al., 2020) and TLogic (Liu et al., 2022), aims to capture causal associations among events in TKGs. The main idea is walking TKGs using rules, where a walking path forms a inference chain, eventually reaching the final entity. This walking method necessitates a higher level of completeness in the inference chain within TKGs, leading to inference breaks when key cue events related to the final entity are missing.

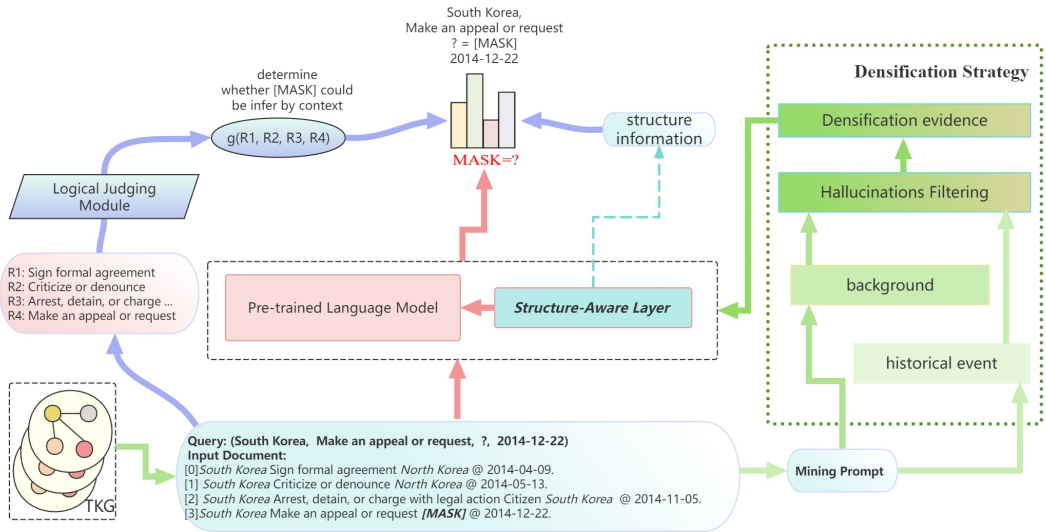

South Korea,
Make an appeal or request
? = [MASK]
2014-12-22
determine whether [MASK] could be infer by context
g(R1, R2, R3, R4)
MASK=?
structure information
Densification Strategy
Densification evidence
Hallucinations Filtering
background
historical event
Logical Judging Module
R1: Sign formal agreement
R2: Criticize or denounce
R3: Arrest, detain, or charge ...
R4: Make an appeal or request
Pre-trained Language Model
Structure-Aware Layer
TKG
Query: (South Korea, Make an appeal or request, ?, 2014-12-22)
Input Document:
[0]South Korea Sign formal agreement North Korea @ 2014-04-09.
[1] South Korea Criticize or denounce North Korea @ 2014-05-13.
[2] South Korea Arrest, detain, or charge with legal action Citizen South Korea @ 2014-11-05.
[3]South Korea Make an appeal or request [MASK] @ 2014-12-22.
Mining Prompt

Figure 2: SALMON overview: a PLM-based framework with a *structure-aware layer* inside, a *logical judging module*, and a *densification strategy*.

2.3 LMs-based models

KG-BERT (Yao et al., 2019) and LASS (Shen et al., 2022) represent two traditional *non-temporal* KG reasoning methods. The latest approach based on pre-trained language models (PLMs) in *temporal* knowledge graph reasoning is PPT (Xu et al., 2023). By transforming sampled quadruples into PLM's inputs and employing a masking strategy, PPT effectively integrates temporal knowledge graph information into language models. However, due to the sparsity of TKGs, many events lack historical information, limiting PPT's ability to fully explore the deeper reasons for associations among events and resulting in performance limitations. ICL (Lee et al., 2023) which is a temporal reasoning model based on large language models (LLMs), exploring the use of in-context learning (ICL) with LLMs for forecasting in TKGs, demonstrating that LLMs can achieve TKGR without additional training by leveraging patterns in the historical context.

3 Problem Formulation

A *temporal knowledge graph* (TKG) is an assembly of facts, symbolized by $G \subset E \times R \times E \times T$, where E, R and T denote the finite sets of entities, relations, and timestamps, respectively. A quadruple $(s,\ r,\ o,\ t)$ describes that a fact of relation type $r \in R$ occurs between the subject entity $s \in E$ and the object entity $o \in E$ at the timestamp $t \in T$. Boldfaced $\boldsymbol{s}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{t}$ represent their embeddings.

The goal of the *TKG reasoning* (TKGR) task is to predict the missing object entity o via answering query like $q = (s_q,\ r_q,\ ?,\ t_q)$ with only historical known facts $\{(s,\ r,\ o,\ t_i)|t_i < t_q\}$ given. Not that, without loss of generality, when predicting the missing subject of a query $q = (?,\ r_q,\ o_q,\ t_q)$, we can convert the query into $q = (o_q,\ r_q^{-1},\ ?,\ t_q)$.

4 Methodology

4.1 Overview

SALMON is built upon a PLM-based framework as illustrated in Figure 2, incorporating a dedicated *structure-aware layer* to jointly capture evolving patterns and structural information within TKGs. A *logical judging module* is also inserted into the learning process to identify which events in the evolving pattern are closely related, further enhancing the model's reasoning capabilities. Additionally, a *densification strategy* based on LLMs is proposed. This strategy seeks to enhance the model's inference abilities by mining and infusing necessary latent knowledge for TKGR, especially for reasoning tasks lacking historical information, thereby augmenting the prediction accuracy.

4.2 Structure-Aware Layer

In order to capture both temporal evolution and structural characteristics, we design a structure-aware layer which could be easily incorporated into Transformer-based language model.

Our model SALMON contains L identical blocks, each of which is composed sequentially of the following modules: multi-head self-attention

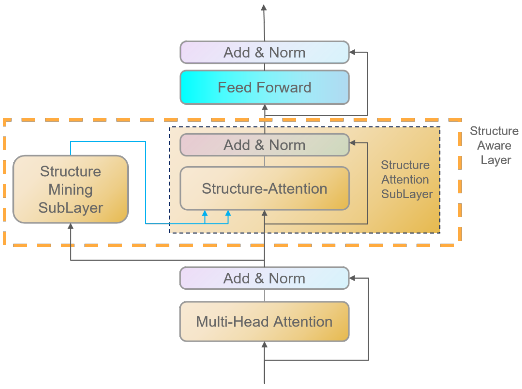


Figure 3: Structure-Aware Transformer Block.

layer, ***structural-aware layer***, feed forward layer, and several layer-normalization layers, as in Figure 3. Note that the data input to our model is a concatenation of a finite number of quadruples, which is same as PPT (Xu et al., 2023). Next, we detail the core layer of SALMON, i.e., structural-aware layer consisting of two sublayers: *structure-mining sublayer*, *structure-attention sublayer*.

Structure-Mining Sublayer This sublayer is dedicated to mining structural information for subsequent integration in the structural attention layer. We denote the input to this sub-layer as $[\boldsymbol{x_1}, ..., \boldsymbol{x_n}]$ where $\boldsymbol{x_i} \in \mathbb{R}^d$. Specifically, the sublayer calculates the corresponding key $\boldsymbol{k}$ and value $\boldsymbol{v}$ required by the upcoming structural attention layer. As the result of attention is $\mathbb{E}_{x \sim k|q}[v(x)]$, we need to incorporate structure information into $\boldsymbol{v}$. We make two *adjustments* to $\boldsymbol{v}$ as depicted in Figure 4: (*i*) If x_i corresponds to a subject s in the input sequence, we add the embedding of entities that share the same relation r as s, considering these entities to have similar behaviors to s; (*ii*) If x_i corresponds to an object, we add embedding of historical relations between the subject and the object, which is modeled as the average-pooled embedding of the relations. The adjustments can be formulated as:

$$\boldsymbol{v_i} = \begin{cases} W^V \boldsymbol{x_i} + W^{Vs} S_{x_i} H_E, & x_i \in sub \\ W^V \boldsymbol{x_i} + W^{Vr} R_{x_i} H_R, & x_i \in obj \end{cases} \tag{1}$$

$$\boldsymbol{k_i} = W^K \boldsymbol{x_i} \tag{2}$$

where $W^V \in \mathbb{R}^{d \times d}$, $W^K \in \mathbb{R}^{d \times d}$, $W^{Vs} \in \mathbb{R}^{d \times d}$ and $W^{Vr} \in \mathbb{R}^{d \times d}$ are projection matrices , H_E and H_R are embedding matrices of the entity and the relation, $S_{x_i} \in \mathbb{R}^{|E|}$ denotes which entity has similar behavior to s_q, $R_{x_i} \in \mathbb{R}^{|R|}$ denotes which relation also occurs between s_q and x_i.

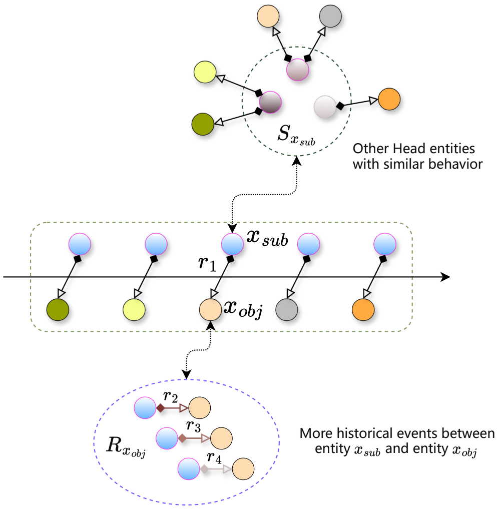

$S_{x_{sub}}$
Other Head entities with similar behavior
x_{sub}
r_1
x_{obj}
r_2
r_3
r_4
$R_{x_{obj}}$
More historical events between entity x_{sub} and entity x_{obj}

Figure 4: Enhancing the incorporation of structural information: Injecting information about entities with similar behaviors into the head entity, and injecting more historical interaction information between the head entity and the tail entity.

Our design above is based on the following idea: when predicting the trends of a specific entity s_q, it is crucial to refer to other entities that exhibit similar behaviors. For instance, to predict the behavior of Donald Trump, who frequently participates in various activities including signing agreements or engaging in negotiations. We can consider the behavior of other entities, such as Barack Obama, who also frequently signs agreements or engages in negotiations. Additionally, to incorporate additional structural information, we introduce additional context into the tail entity, revealing other events related to the head entity. This incorporation of structural information is depicted in Figure 4.

Structure-Attention Sublayer This sublayer, a variant of multi-head cross-attention with twelve attention heads. The first four attention heads are configured with masks to constrain their attention exclusively on subjects, ensuring that the attention mechanism captures shared structural features among similar entities within the input sequence. In contrast, the subsequent four attention heads are also equipped with masks to restrict their attention solely to objects, facilitating the extraction of global structural information related to tail entities in the input sequence. And we keep the last four attention heads the same as normal attention heads. The process for the first four attention heads can be formulated as follows:

$$\boldsymbol{a_i} = Softmax[\frac{(\boldsymbol{x_i} W^Q)\,[\boldsymbol{k_1}, .., \boldsymbol{k_n}]}{\sqrt{d_k}} + Mask(k_j \notin sub)] \tag{3}$$

$$\boldsymbol{z_i} = \textstyle\sum_{j=1}^{n} a_{ij} \boldsymbol{v_j} \tag{4}$$

where $W^Q \in \mathbb{R}^{d \times d}$ is a trainable projection matrix, $\boldsymbol{a_i} \in \mathbb{R}^d$ is the attention vector of x_i, and $\boldsymbol{z_i} \in \mathbb{R}^d$ is the output representation of x_i. The next four heads use the same formula, but with $Mask(k_j \notin sub)$ replaced by $Mask(k_j \notin obj)$.

4.3 Logicality Judging Module

To further enhance the model's reasoning capabilities, we design a logical judging module. This module is designed to give priority to learning the most relevant evolving information of logical causal associations in TKGs during the training process of language models. By incorporating logicality into the learning process, SALMON is better equipped to make informed predictions and decisions, contributing to the overall effectiveness of TKGR.

This module draws inspiration from BERT (Kenton and Toutanova, 2019), which presents a groundbreaking approach to language representation learning via a training objective known as the "masked language model". To delve into the details of this module, consider a set of discrete tokens represented as $x = \{x_1, x_2, ..., x_L\}$, where each x_i belongs to the token vocabulary X. The model defines a joint probability distribution over this token set as follows:

$$p(x|\theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{L} \varphi_i(x|\theta) \propto \exp\left(\sum_{i=1}^{L} \log \varphi_i(x|\theta)\right) \tag{5}$$

where φ_i represents the i-th potential function characterized by parameters θ, while Z denotes the partition function. The log potential (energy) functions for each location are defined by

$$P = log\ \varphi_i(x|\theta) = \boldsymbol{x_i}^T f_\theta(x_{\setminus i}) \tag{6}$$

$$x_{\setminus i} = (x_1,\ ..., x_{i-1}, [MASK], x_i,\ ..., x_L) \tag{7}$$

where the function $f_\theta(x_{\setminus i})$ is a multi-layer bidirectional transformer model.

We identify a limitation in the aforementioned formula: it does not consider whether the current $[MASK]$ can be genuinely inferred from the context. In certain situations, the current $[MASK]$ and the context are independent, but optimizer will still force model to fit a conditional distribution, capturing illogical errors in contextual associations.

To alleviate this problem, we innovatively design a *new rule-based log potential functio*n P_{rule}:

$$\begin{aligned}
&log\ P_{rule} \\
&= g(q, G_{1:t_q-1})\ log[\frac{exp(o_q^T f(q|G_{1:t_q-1}))}{\sum_{o_i \in E} exp(o_i^T f(q|G_{1:t_q-1}))}] \\
&\approx g[e, e_1, ...,\ e_n]\ log[\frac{exp(o_q^T f(q|G_{1:t_q-1}))}{\sum_{o_i \in E} exp(o_i^T f(q|G_{1:t_q-1}))}] \\
&\approx g[e, e_1, ...,\ e_n]\ h_{o_q}
\end{aligned} \tag{8}$$

where o_q represents the embedding of the answer to the query, E denotes the finite set of entities, $e_1, ..., e_n$ represent events in the evolution line, q represents the current query $(s, p, ?, t)$. For simplicity, we denote the second term of Eq.(8) as h_{o_q}.

Here, the function g above represents the strength of the causal relationship between the result of current query and the context in an evolving line. It will help to refine the model's decision-making process, prioritizing the most relevant information for reasoning. We posit that the strength of causality is determined by the interconnection of events, and furthermore, this interconnection is largely independent of entities. So we formulate g as follows:

$$\begin{aligned}
g[e, e_1, ...,\ e_n] &\doteq g[r_e, r_{e_1}, ...,\ r_{e_n}] \\
&\approx M\,[\alpha(r_e, r_{e_1}), ..., \alpha(r_e, r_{e_n})]
\end{aligned} \tag{9}$$

where the function α stands for the relativity of two relations, meanwhile the function M stands for the merge operation on all relativity. Based on this idea, we define α as the co-occurrence confidence between relations, as shown in Eq.(10), where the numerator represents the number of entity pairs (s, o) that have both relations r_{e_x} and r_{e_y}, and the denominator represents the number of entity pairs (s, o) that have relation r_{e_x}.

$$C(r_{e_x}, r_{e_y}) = \frac{\#(s, o) : (s, r_{e_x}, o, t_i) \wedge (s, r_{e_y}, o, t_j)}{\#(s, o) : (s, r_{e_x}, o, t_x)} \tag{10}$$

Considering that the function α do most heavy lifting, so for simplicity, we set the Function M as $AVERAGE$. Thus, with the help of logicality judging module, the most qualified evolution sequence information will be prioritized for training.

4.4 Densification Strategy

In TKGs, some events have very limited related historical events. In order to alleviate the issue, we aim at mining evidence necessary for reasoning. With the help of sufficient evidence, we believe that the prediction would be much more credible and accurate. Given that LLMs possess substantial knowledge reserves, we propose a densification strategy based on LLMs specifically for TKGR.

Our densification strategy consists of three steps. The *first step* involves two mining prompts, one focuses on background information and the other on historical events. Given that large models are prone to hallucinations, the first step may yield evidence containing errors. Therefore, as a *second step*, we leverage the large model once more to scrutinize and filter out any evidently unreasonable data. In the *third step*, we inject the generated evidence corpus into the model.

Step-1: Mining Evidence From Multiple View

Background-Mining Prompt T_B is to extract relevant background information about the entities in the quadruple. To achieve this, we carefully design a prompt template, which consists of three parts. The first part outlines the task, specifying how the LLM should gradually contemplate and generate evidence. The second part provides a demonstration, offering the LLM with insights into what constitutes valid evidence. The third part includes the quadruple for which background information is currently being mined.

Historical-Event-Mining Prompt T_H is to explore more historical events that occurred between the head/tail entities within a quadruple, or events related to one of the entities. To accomplish this objective, we meticulously crafted a prompt, structured into three components, which is similar to the Background-Mining Prompt.

Step-2: Filter Unreasonable Evidence

The step-1 may generate flawed evidence due to the LLM's hallucinatory output issue. Consequently, we further utilize the LLMs again to meticulously examine and eliminate any obviously implausible evidence through our designed Hallucinations Flitering Template (T_F).

The details of three prompts (T_B, T_H, T_F) can be found in Appendix A.1.

Step-3: Implicit Densification

After the first two steps, we obtain an Evidence Corpus $\mathcal{EC}$ as showcased in Appendix A.2. We employ an implicit densification approach to seamlessly integrate the $\mathcal{EC}$ into the SALMON. We achieve this by minimizing the SALMON's Masked Language Model (MLM) loss on the $\mathcal{EC}$.

The above process of densification strategy is summarized in the Algorithm of Appendix A.3.

4.5 Training

The training process of SALMON is divided into *two stages*. In the first stage, we perform densification strategy on the training set to obtain the Evidence Corpus $\mathcal{EC}$. We subsequently train a PLM by minimizing its MLM loss on the $\mathcal{EC}$. In the second stage, we first sample a series of same-head quadruples on the training set and concatenate them into evolving lines, which are then aggregated to form an evolving corpus as shown previously in Figure 2. Next, we integrate our Structure-Aware Layer into the PLM trained in the first stage and introduce the Logicality Judging Module to enhance the training process. Ultimately, this results in the final SALMON model. The formulas involved in this process can be found in Appendix D.

5 Experiment

5.1 Experiment Setup

Datasets and Evaluation Metrics We conduct experiments on three commonly used benchmark datasets for TKGR, including ICEWS14 (García-Durán et al., 2018), ICEWS18 (Boschee et al., 2015), and ICEWS0515 (García-Durán et al., 2018). We compute the mean reciprocal rank (MRR) and hits@k for $k \in \{1, 3, 10\}$. The statistics of datasets, the evaluation metrics and settings are detailed in Appendix B.

Baseline methods In addition to selecting the existing non-temporal models, e.g., DistMult (Yang et al., 2015) and others (Xu et al., 2023). We mainly compare SALMON with the state-of-the-art temporal baselines, including:

- **Embedding/Rule-based Temporal Models**: HyTE (Dasgupta et al., 2018), TTransE (Jiang et al., 2016), TA-DistMult (García-Durán et al., 2018), RGCRN (Seo et al., 2018), CyGNet (Zhu et al., 2021), RE-NET (Jin et al., 2019), RE-GCN (Li et al., 2021), and the rule-based TKGR model TLogic (Liu et al., 2022).

- **LMs-based Temporal Models**: PPT (Xu et al., 2023) is the previous SOTA TKGR model based on PLMs, and ICL (Lee et al., 2023) is a LLM-based TKGR method.

	ICEWS18				**ICEWS05-15**				**ICEWS14**			
Method	MRR	Hits@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
DistMult	13.86	5.61	15.22	31.26	19.91	5.63	27.22	47.33	20.32	6.13	27.59	46.61
ComplEx	15.45	8.04	17.19	30.73	20.26	6.66	26.43	47.31	22.61	9.88	28.93	47.57
R-GCN	15.05	8.13	16.49	29.00	27.13	18.83	30.41	43.16	28.03	19.42	31.95	44.83
ConvE	22.81	13.63	25.83	41.43	31.40	21.56	35.70	50.96	30.30	21.30	34.42	47.89
ConvTransE	23.22	14.26	26.13	41.34	30.28	20.79	33.80	49.95	31.50	22.46	34.98	50.03
RotatE	14.53	6.47	15.78	31.86	19.01	10.42	21.35	36.92	25.71	16.41	29.01	45.16
HyTE	7.41	3.10	7.33	16.01	16.05	6.53	20.20	34.72	16.78	2.13	24.84	43.94
TTransE	8.44	1.85	8.95	22.38	16.53	5.51	20.77	39.26	12.86	3.14	15.72	33.65
TA-DistMult	16.42	8.60	18.13	32.51	27.51	17.57	31.46	47.32	26.22	16.83	29.72	45.23
RGCRN	23.46	14.24	26.62	41.96	35.93	26.23	40.02	54.63	33.31	24.08	36.55	51.54
CyGNet	26.46	16.62	30.57	45.58	35.46	25.44	40.20	54.47	35.45	26.05	39.91	53.20
RE-NET	26.17	16.43	29.89	44.37	36.86	26.24	41.85	57.60	35.77	25.99	40.10	54.87
RE-GCN	<u>27.51</u>	<u>17.82</u>	<u>31.17</u>	<u>46.55</u>	38.27	27.43	43.06	**59.93**	37.78	27.17	42.50	**58.84**
Tlogic[†]	-	15.50	27.20	41.20	-	-	-	-	-	26.50	39.50	53.10
ICL[†]	-	13.60	22.40	32.10	-	-	-	-	-	24.70	36.30	47.10
PPT	26.63	16.94	30.64	45.43	<u>38.85</u>	<u>28.57</u>	<u>43.35</u>	58.63	<u>38.42</u>	**28.94**	<u>42.50</u>	57.01
SALMON	**28.57**	**18.75**	**32.66**	**47.70**	**39.38**	**29.10**	**43.95**	<u>59.37</u>	**38.78**	<u>28.92</u>	**43.29**	<u>57.83</u>
APG	1.06	0.93	1.49	1.15	0.53	0.53	0.60	-0.56	0.36	-0.02	0.79	-1.01
RPG (%)	3.85	5.22	4.78	2.47	1.36	1.86	1.38	-0.93	0.94	-0.07	1.86	-1.72

Table 1: Comparison between our SALMON and baselines. APG and RPG indicate the absolute performance gains and the relative performance gains achieved by our model compared with the best-performing baselines (REGCN or PPT). APG and RPG can be calculated by $APG = R_{ours} - R_{baseline}$ and $RPG = (R_{ours} - R_{baseline})/R_{baseline}$, where R_{ours} and $R_{baseline}$ denote the results of our model and baselines (REGCN or PPT), respectively. Best results are in bold, and the second best are underlined. The results indicated by † are reported in Lee et al. (2023), and the results of other baselines are referred from Xu et al. (2023).

5.2 Experiment Results

We report the results of SALMON and baselines in Table 1. The results show that SALMON outperforms baselines across all most of settings on three datasets.

1. Compared with the performance of the non-temporal models, our model SALMON outperforms all baselines (average of **6.87** improvement in MRR), showing that SALMON can better capture the temporal dependencies in TKGs.

2. Compared with embedding-based temporal models, our model is also better than most of baselines, only RE-GCN performs better than SALMON for hits@10 on the datasets ICEWS14 and ICEWS05-15. The results show that our model has stronger temporal representation and reasoning abilities. Moreover, compared with the rule-based model Tlogic, our SALMON achieves significant average improvements of up to **2.84** Hits@1, **4.63** Hits@3, **5.62** Hits@10 across two datasets, demonstrating the effectiveness of our rules.

3. Compared with the PLM-based model, our SALMON are better than the previous SOTA model PPT on almost all metrics (except that it is slightly lower than PPT in Hits@1 on ICEWS14). SALMON improves MRR value by about **1.94** on the ICEWS18 dataset. The overall better performance demonstrates that SALMON further enhances PLM's understanding of TKGs.

4. Our model SALMON also outperforms the LLM-based model ICL in all metrics. It is observed that **5.15** and **4.22** Hits@1 improvements on ICEWS18 and ICEWS14 datasets, which show that SALMON better fits the reasoning in TKGs by training a small language model and assisting with a LLM-based implicit knowledge densification strategy.

5.3 Ablation Study

We conduct ablation studies to understand the contribution of different model components of SALMON as shown in Table 2.

Impact of Structure-Aware Layer We insert a structure-aware layer into the SALMON model. When the structure-aware layer is removed (-SAL), there is a decline in performance metrics. Such change causes 0.22 drop in terms of Hits@3 on ICEWS18. Similar trends are observed in other metrics, demonstrating the significant contribution of the structure-aware layer.

Method	ICEWS18 MRR	ICEWS18 Hits@1	ICEWS18 Hit@3	ICEWS18 Hit@10	ICEWS05-15 MRR	ICEWS05-15 Hit@1	ICEWS05-15 Hit@3	ICEWS05-15 Hit@10	ICEWS14 MRR	ICEWS14 Hit@1	ICEWS14 Hit@3	ICEWS14 Hit@10
SALMON	**28.57**	**18.75**	**32.66**	**47.70**	**39.38**	29.10	**43.95**	59.37	**38.78**	**28.92**	**43.29**	57.83
- Logic - Dens - SAL	26.63	16.94	30.64	45.43	38.85	28.57	43.35	58.63	38.05	28.45	42.40	56.27
- Logic - SAL	26.76	16.92	30.51	46.17	38.95	28.61	43.58	58.89	38.08	28.15	42.91	56.92
- Logic - Dens	26.91	17.03	30.98	46.31	38.98	28.77	43.29	58.94	38.06	27.71	42.69	**58.33**
- Dens - SAL	27.36	17.42	31.36	46.82	39.01	28.62	43.99	58.99	38.17	28.31	42.28	57.59
- Logic	27.91	17.85	32.03	47.34	39.28	28.62	44.37	**59.96**	38.38	28.54	42.70	57.63
- Dens	28.26	18.33	32.44	47.38	39.22	29.25	43.66	58.58	38.27	28.24	42.61	57.50
- SAL	28.48	18.63	32.40	47.67	39.34	**29.13**	43.84	59.25	38.46	28.55	43.12	57.21

Table 2: Ablation Study on eliminating the structure-aware layer (-SAL), logical judging module (-Logic) and densification strategy (-Dens) from SALMON.

Impact of Logical Judging Module The SALMON model, with the logical judging module included, generally exhibits superior performance across various metrics. For instance, on the ICEWS18 dataset, the full model achieves an MRR of 28.57 and Hits@1 of 18.75. Removing the logical judging module (-Logic) leads to a noticeable drop in MRR to 27.91 and Hits@1 to 17.85, underscoring the module's contribution.

While the logical judging module significantly improves precision at higher ranks (e.g., Hits@1), it may slightly impact broader retrieval accuracy (e.g., Hits@10). For instance, on the ICEWS05-15 dataset, while Hits@1 increases with the module, Hits@10 slightly decreases.

Impact of Densification Strategy The inclusion of the densification strategy in SALMON significantly improves overall performance metrics. Removing the densification strategy (-Dens) causes 0.31 and 0.42 drop in terms of MRR and Hits@1 on ICEWS18 respectively. More concretely, we can get more observations from the results of deleting different combination modules in Table 2, which further suggest that the structure-aware layer, logical judging module and densification strategy are all important.

5.4 Analysis of the Intrinsic TKGR Capability of the LLM used for Densification

Since our method injects necessary evidence mined by the LLM via the densification strategy, we conduct further experimental analysis to verify whether the LLM itself has the ability to directly use these information to effectively implement TKGR.

We conduct experiments on the TKGR task under four settings (detailed in Appendix C), based on the LLM Llama2-7b-chat previously used for densification strategy. The results are shown in Table 3. we observe that the direct use of the LLM exhibits weak temporal reasoning capabilities, often yielding incorrect answers or refusing to respond to a majority of queries. However, these results may also indirectly show that the gain of our model does not come entirely from the evidence information mined by the LLM, but is based on the structure-aware layer, logical judging module and densification strategy we proposed.

LLM Setting	ICEWS18 Acc	ICEWS18 $Refuse$	ICEWS18 $Wrong$
unsupervised	0.01	0.63	0.36
unsupervised+cot	0.03	0.25	0.72
supervised	0.07	0.58	0.35
supervised+cot	0.08	0.20	0.72

Table 3: Performance of using the LLM directly for TKGR. Acc, $Wrong$, $Refuse$ denotes the frequency of answer accurately, incorrectly and refusal to answer, respectively.

LLM	**MRR**	**hit@1**	**hit@3**	**hit@10**
Llama-2-7b-chat	28.57	18.75	32.66	47.74
Llama-3-8B-Instruct	28.65	18.74	32.89	47.84
Qwen2-7B-Instruct	28.44	18.59	32.44	47.56

Table 4: Comparison of different LLMs in the densification strategy.

5.5 Effectiveness of Different LLMs in Densification Strategy

To assess the effectiveness and universality of the densification strategy, we evaluated the impact of incorporating various large language models (LLMs). Specifically, we integrated two additional LLMs—Qwen2-7B-Instruct and Llama3-8B-Instruct—into the densification process alongside the original Llama-2-7b-chat used in the SALMON model. The results on the ICEWS18 dataset are summarized in Table 4.

	Sparse ICEWS18				ICEWS18			
	MRR	hit@1	hit@3	hit@10	MRR	hit@1	hit@3	hit@10
SALMON (-Dens)	23.01	14.19	26.84	39.93	28.26	18.33	32.44	47.38
SALMON	24.69	15.55	28.49	42.21	28.57	18.75	32.66	47.70
Dense Gain	**1.67**	**1.36**	**1.65**	**2.28**	0.31	0.42	0.22	0.32
Dense Gain (%)	**7.25%**	**9.58%**	**6.14%**	**5.70%**	1.09%	2.29%	0.67%	0.48%

Table 5: Experiments on how the sparsity of TKG specifically affects model performance and how the densification strategy can mitigate these effects, where "-Dens" denotes our SALMON model without the densification strategy.

The results indicate that: (*i*) Incorporating different LLMs within the densification strategy leads to consistently good performance, highlighting the robustness of our method. (*ii*) Utilizing models with more parameters, such as Llama-3-8B-Instruct, demonstrate slightly better performance, suggesting that the densification strategy has significant potential to improve further with stronger LLMs. These results support the versatility and potential of our densification strategy across various LLMs.

5.6 Robustness of Densification Strategy under extremely Sparse Situations

To provide a quantitative assessment of how the sparsity of TKGs affects model performance and to evaluate the effectiveness of our densification strategy in highly sparse conditions, we performed additional experiments. Specifically, we created a more sparse TKG by retaining only 10% of the original outgoing edges per entity (with a minimum of 1 edge per entity). We then applied our densification strategy to this more sparse TKG and compared its performance against the SALMON model both with and without the densification strategy.

The results in Table 5 demonstrate that increased sparsity significantly diminishes model performance. The MRR drops from 28.57 to 24.69. This trend is consistent across other evaluation metrics such as hit@1, hit@3, and hit@10. Importantly, the densification strategy effectively counteracts these performance losses, achieving a notable improvement in MRR by 1.67 points when applied to the sparse TKG. In contrast, the improvement in MRR for the original TKG was 0.31 points.

The gain percentages of the densification strategy on the original TKG are [1.09%, 2.29%, 0.67%, 0.48%] for [MRR, hit@1, hit@3, hit@10], respectively. When applied to the sparse TKG, these gains increase to [7.25%, 9.58%, 6.14%, 5.70%] for the same metrics. These findings demonstrate that our densification strategy not only improves model performance but also maintains robustness in extremely sparse scenarios.

6 Conclusion

In this paper, we propose a novelty TKGR model SALMON. Specifically, we design a PLM-based framework with a structure-aware layer inside to jointly capture evolving patterns and structural information in TKGs. Moreover, we propose a logical judging module to give priority to learning the most relevant evolving information in TKGs during the training process. Furthermore, we propose a LLMs-based densification strategy to alleviate the difficulty of the model in making accurate inferences about sparse events in TKGs. Results on TKGR benchmarks demonstrate the effectiveness of our method.

Limitations

Explanation limitations. Although SALMON introduces a logical module to improve the logic of reasoning, the decision-making process of the model may still not be transparent. Because PLM, the base of SALMON, is a black box model.
LLM selection limitation in densification strategy. We have not fully tried the use of other LLMs (e.g., GPT-4 and others). Existing LLM methods (e.g., ICL (Lee et al., 2023) and GenTKG (Liao et al., 2024)) have already achieved good results. In subsequent work, we will further explore how to efficiently use other LLMs to enhance the performance of SALMON.

Acknowledgments. The authors sincerely thank the reviewers for their valuable comments, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057), and Sponsored by CAAI-MindSpore Open Fund, developed on OpenI Community.

References

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12:2.

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

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2001–2011.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4816–4821.

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

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37.

Prachi Jain, Sushant Rathi, Soumen Chakrabarti, et al. 2020. Temporal knowledge base completion: new algorithms and evaluation protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3733–3747.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Encoding temporal information for time-aware link prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2350–2354.

Woojeong Jin, Changlin Zhang, Pedro Szekely, and Xiang Ren. 2019. Recurrent event network for reasoning over temporal knowledge graphs. In *International Conference on Learning Representations (ICLR)*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *International Conference on Learning Representations (ICLR)*.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, pages 544–557.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *SIGIR*, pages 408–417.

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

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.

Zhengtao Liu, Lei Tan, Mengfan Li, Yao Wan, Hai Jin, and Xuanhua Shi. 2023. Simfy: A simple yet effective approach for temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3825–3836.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, pages 362–373.

Jianhao Shen, Chenguang Wang, Linyuan Gong, and Dawn Song. 2022. Joint language semantic and structure embedding for knowledge graph completion. *arXiv preprint arXiv:2209.08721*.

Xing Tang and Ling Chen. 2023. Gtrl: An entity group-aware temporal knowledge graph representation learning method. *arXiv preprint arXiv:2302.11091*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, pages 2071–2080. PMLR.

Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 5730–5746.

Chengjin Xu, Yung-Yu Chen, Mojtaba Nayyeri, and Jens Lehmann. 2021. Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2569–2578.

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

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

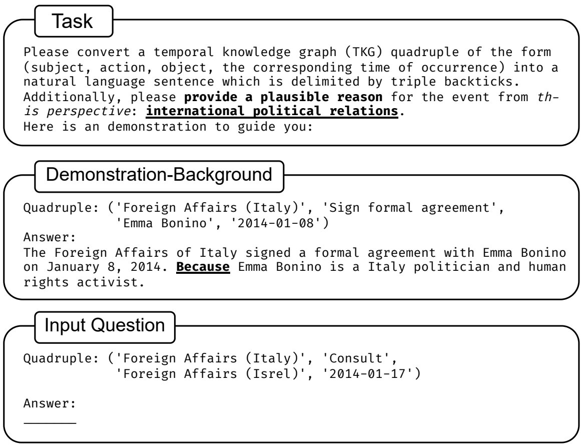
Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4732–4740.

A Densification Strategy

A.1 Details of Prompt Templates

Details of our mining prompt template (T_B, T_H) are shown in Figure 5 and Figure 6.



Task

Please convert a temporal knowledge graph (TKG) quadruple of the form (subject, action, object, the corresponding time of occurrence) into a natural language sentence which is delimited by triple backticks.
Additionally, please **provide a plausible reason** for the event from *this perspective*: **international political relations**.
Here is an demonstration to guide you:

Demonstration-Background

Quadruple: ('Foreign Affairs (Italy)', 'Sign formal agreement',
'Emma Bonino', '2014-01-08')
Answer:
The Foreign Affairs of Italy signed a formal agreement with Emma Bonino on January 8, 2014. **Because** Emma Bonino is a Italy politician and human rights activist.

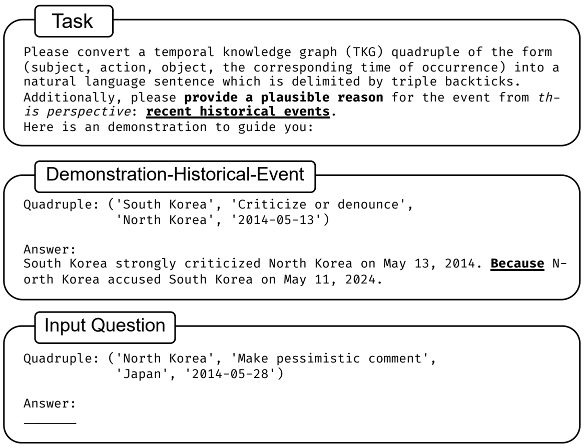
Input Question

Quadruple: ('Foreign Affairs (Italy)', 'Consult',
'Foreign Affairs (Isrel)', '2014-01-17')

Answer:
______

Figure 5: Background-Mining Prompt Template

The hallucinations flitering template T_F is shown in Figure 7. The first part of the template involves task specification based on role-playing, where we prompt the LLM to play the role of an expert with extensive historical knowledge. Subsequently, the model is tasked with determining whether a given event is Reasonable or Unreasonable. The second part focuses on a demonstration with balanced positive and negative examples. Balancing the categories aims to prevent the model from favoring the class with more samples. We carefully select evidence for both categories, choosing a markedly flawed example for Unreasonable and a moderately reliable one for Reasonable. Through this selection process, we find that the LLM can effectively filter data. The third part presents the evidence that is currently being judged as Reasonable or Unreasonable.



Task

Please convert a temporal knowledge graph (TKG) quadruple of the form (subject, action, object, the corresponding time of occurrence) into a natural language sentence which is delimited by triple backticks.
Additionally, please **provide a plausible reason** for the event from *this perspective*: **recent historical events**.
Here is an demonstration to guide you:

Demonstration-Historical-Event

Quadruple: ('South Korea', 'Criticize or denounce',
'North Korea', '2014-05-13')

Answer:
South Korea strongly criticized North Korea on May 13, 2014. **Because** North Korea accused South Korea on May 11, 2024.

Input Question

Quadruple: ('North Korea', 'Make pessimistic comment',
'Japan', '2014-05-28')

Answer:
______

Figure 6: Historical Event Mining Prompt Template

A.2 Evidence Corpus Showcase

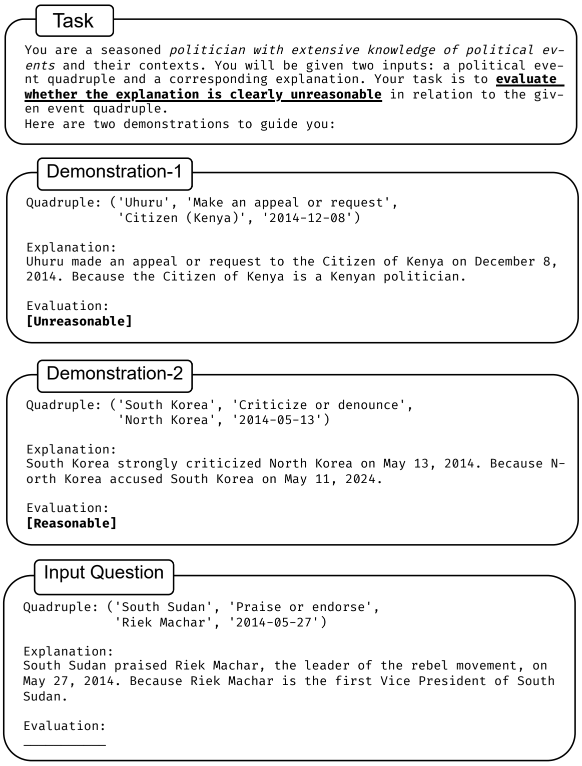
We select some evidence in Evidence Corpus $\mathcal{EC}$ to demonstrate the effectiveness of the densification strategy, as shown in Figure 8.

A.3 Algorithm of Densification Strategy

The pseudocode for Densification Strategy is shown in Algorithm 1.

B Datasets and Experimental Settings

We conduct experiments on the dataset Integrated Crisis Early Warning System (ICEWS) (Boschee et al., 2015), which contains information about international events and is a commonly used benchmark dataset for link prediction on TKGs. We choose the subsets ICEWS14 (García-Durán et al., 2018), ICEWS18 (Boschee et al., 2015), and ICEWS0515 (García-Durán et al., 2018), which include data from the years 2014, 2018, and 2005 to 2015, respectively. Note that each dataset is split into training, validation, and test set, so that the



Task

You are a seasoned *politician with extensive knowledge of political events* and their contexts. You will be given two inputs: a political event quadruple and a corresponding explanation. Your task is to **evaluate whether the explanation is clearly unreasonable** in relation to the given event quadruple.
Here are two demonstrations to guide you:

Demonstration-1

Quadruple: ('Uhuru', 'Make an appeal or request',
'Citizen (Kenya)', '2014-12-08')

Explanation:
Uhuru made an appeal or request to the Citizen of Kenya on December 8, 2014. Because the Citizen of Kenya is a Kenyan politician.

Evaluation:
[Unreasonable]

Demonstration-2

Quadruple: ('South Korea', 'Criticize or denounce',
'North Korea', '2014-05-13')

Explanation:
South Korea strongly criticized North Korea on May 13, 2014. Because North Korea accused South Korea on May 11, 2024.

Evaluation:
[Reasonable]

Input Question

Quadruple: ('South Sudan', 'Praise or endorse',
'Riek Machar', '2014-05-27')

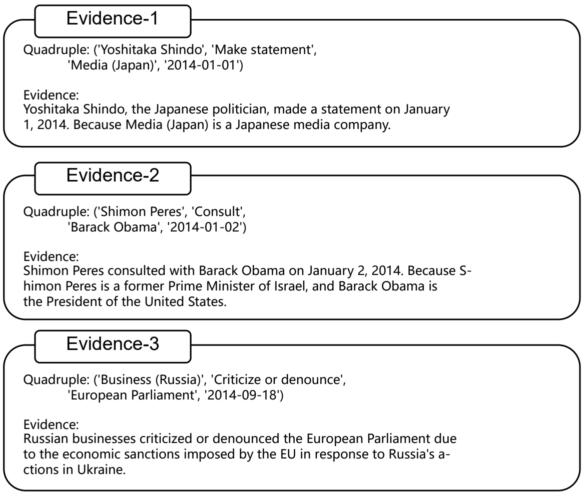
Explanation:
South Sudan praised Riek Machar, the leader of the rebel movement, on May 27, 2014. Because Riek Machar is the first Vice President of South Sudan.

Evaluation:
____

Figure 7: Hallucinations Flitering Template

Algorithm 1: Densification Strategy

```
Input:
$\mathcal{G}$: Training set of TKG
$\mathcal{M}_g$: The LLM for generating evidences
$\mathcal{M}_f$: The LLM for filtering hallucinations
$T_B, T_H, T_F$: Prompt Template for Background Mining, Historical Event Mining, Hallucinations Flitering
Output: $\mathcal{EC}$: Evidence Corpus
$\mathcal{EC} \leftarrow \emptyset$
for quadruple $q \in \mathcal{G}$ do
    // Step 1: Generate evidence
    Construct Prompt $p_b$ by $T_B$ and $q$ ;
    Generate evidence from $\mathcal{G}$ using $\mathcal{M}_g$;
    $Ev_b \leftarrow \mathrm{M}_g(p_b)$
    Construct Prompt $p_h$ by $T_H$ and $q$ ;
    Generate evidence from $\mathcal{G}$ using $\mathcal{M}_g$;
    $Ev_h \leftarrow \mathrm{M}_g(p_h)$
    // Step 2: Filter hallucinations
    for evidence $ev \in \{Ev_b, Ev_h\}$ do
        Construct Prompt $p_f$ by $T_F$ and $ev$ ;
        $Is\_Reasonable \leftarrow \mathrm{M}_f(p_f)$ ;
        if $Is\_Reasonable$ is True then
            $\mathcal{EC} \leftarrow \mathcal{EC} \cup \{ev\}$
return $\mathcal{EC}$
```



Evidence-1

Quadruple: ('Yoshitaka Shindo', 'Make statement',
'Media (Japan)', '2014-01-01')

Evidence:
Yoshitaka Shindo, the Japanese politician, made a statement on January 1, 2014. Because Media (Japan) is a Japanese media company.

Evidence-2

Quadruple: ('Shimon Peres', 'Consult',
'Barack Obama', '2014-01-02')

Evidence:
Shimon Peres consulted with Barack Obama on January 2, 2014. Because Shimon Peres is a former Prime Minister of Israel, and Barack Obama is the President of the United States.

Evidence-3

Quadruple: ('Business (Russia)', 'Criticize or denounce',
'European Parliament', '2014-09-18')

Evidence:
Russian businesses criticized or denounced the European Parliament due to the economic sanctions imposed by the EU in response to Russia's actions in Ukraine.

Figure 8: Examples of evidence in Evidence Corpus

Dataset	$\#E$	$\#R$	**#Granularity**	**#Train**	**#Valid**	**#Test**
ICEWS18	23033	256	24 (hours)	373018	45995	49545
ICEWS05-15	10094	251	24 (hours)	368868	46302	46159
ICEWS14	6869	230	24 (hours)	74845	8514	7371

Table 6: Statistics of the datasets. E and R denote the entities and relations.

timestamps in the training set occur earlier than the timestamps in the validation set, which again occur earlier than the timestamps in the test set. The statistics of the datasets are provided in Table 6.

We compute the mean reciprocal rank (MRR) and hits@k for $k \in \{1, 3, 10\}$. For a rank $x \in \mathbb{N}$, the reciprocal rank is defined as $\frac{1}{x}$, and the MRR is the average of all reciprocal ranks of the correct query answers across all queries. The metric hits@k indicates the proportion of queries for which the correct entity appears under the top k candidates.

The batch size is searched in $\{4, 8, 16, 32\}$ and the learning rate is tuned in $\{1e-4, 3e-4, 5e-4\}$. We use the AdamW optimizer (β_1 = 0.9, β_2 = 0.99). The max epoch of stage-1 training is searched in $\{1, 2, .., 10\}$. We use bert-base-cased as our pre-trained model. We choose an open source LLM Llama2-7b-chat in densification strategy.

C Supplementary Details of LLMs Experiments in Section 5.4

We conduct experiments on the TKGR task under four settings. The first setting *unsupervised* only involves direct inference of queries by the LLM, while the second setting *unsupervised+cot* employs the CoT technique, let the LLM considers relevant evidence before providing an answer. Neither of these two approaches involves training. The *last two settings* represent the supervised counterparts of the aforementioned unsupervised configurations, we utilize the supervised finetuning (SFT) to inject information from the training set into the LLM before inference. To control costs, we sample 100 queries from ICEWS18, repeating the process five times to mitigate sampling bias. We tested four settings based on three evaluation metrics: Acc, $Wrong$, $Refuse$ which stand for the frequency of answer accurately, incorrectly and refusal to Answer.

The prompt template for the *unsupervised* setting and *supervised* setting query LLM is named *Query Template* and is denoted as T_Q, while the prompt template for the latter two setting *unsupervised+cot* and *unsupervised+cot* query LLM is named *Query CoT Template* and is denoted as T_{Q_CoT}.

The details of (T_Q, T_{Q_CoT}) are shown in Table 7 and 8. The Instruction-Tuning Data using in SFT is shown in Table 9. We also present the reasons why LLMs refused to answer in Table 10.

D Formulas involved in the Training Process

We describe the training process in Section 4.5. The formulas involved in the training process are as follows:

$$L_{stage1} = -\sum_{s\in\mathcal{E}\mathcal{C}}\sum_{x\in X_{mask}} logP(x = x_{label}|s) \tag{11}$$

$$\theta^*_{plm} = \underset{\theta_{plm}}{\arg\max}\, L_{stage1} \tag{12}$$

$$\begin{aligned} L_{stage2} &= -\sum_{s\in\mathcal{E}\mathcal{C}}\sum_{e,q\in s} g[e, e_1, ..., \, e_n] \, h_{o_q} \\ &= -\sum_{s\in\mathcal{E}\mathcal{C}}\sum_{e,q\in s} M\left[\alpha(r_e, r_{e_1}), ..., \alpha(r_e, r_{e_n})\right] \, h_{o_q}. \end{aligned} \tag{13}$$

$$\theta_{salmon} = \theta^*_{plm} + \theta_{SAL} \tag{14}$$

$$\theta^*_{salmon} = \underset{\theta_{salmon}}{\arg\min}\, L_{stage2} \tag{15}$$

where θ_{plm} denotes the parameter of the PLM trained in Stage-1, θ_{SAL} denotes the parameter of the Structure-Aware Layer SAL trained in Stage-2. θ_{salmon} denotes the parameter of the full SALMON model in Stage-2.

## Task
You are provided with a quadruple from a temporal knowledge graph. A temporal quadruple typically consists of four components: subject, verb, time, and object. In this case, the subject, verb, and time are given, but the object is missing.
Your task is to infer the most likely object for this quadruple based on the given subject, verb, and time. Remember, this temporal knowledge graph focuses on international relations and political events, so consider these factors in your reasoning.

## Query Quadruple
Subject: {query_subject}
Verb: {query_relation}
Time: {query_time}
Object: ? (Unknown)

## Output Format
Please provide your answer in the following format:
Object: [Your Final Answer Object]

## Output
Generate a plausible answer object for this quadruple:

Table 7: The *unsupervised* querying prompt template in the experiments of Section 5.4. We replace the colored slot with quadruple before querying the LLMs. Note that we use the same template when conducting SFT on LLM.

```
## Task
You are provided with a quadruple from a temporal knowledge graph. A temporal quadruple typically consists of four components: subject, verb, time, and object. In this case, the subject, verb, and time are given, but the object is missing.
Your task is to apply logical reasoning to infer the missing object. Here's a step-by-step approach to guide your thinking:
1. Understand the context: Analyze the subject and verb to identify the type of event or action described. For example, is it a military action, a diplomatic move, or an economic sanction?
2. Consider temporal factors: Take into account the time frame specified in the quadruple. This may help you identify relevant historical events or political developments that could be related to the subject and verb.
3. Apply domain knowledge: ICEWS14 focuses on international relations and political events. Use your knowledge of international politics and current affairs to identify potential candidates for the missing object.
4. Reason logically: Based on the context, temporal factors, and domain knowledge, apply logical reasoning to determine the most plausible object for the quadruple. Consider factors such as causality, precedents, and related events.

## Query Quadruple
Subject: {query_subject}
Verb: {query_relation}
Time: {query_time}
Object: ? (Unknown)

## Output Format
Please provide your answer in the following format:
Thinking: [Your Thinking]
Object: [Your Final Answer Object]

## Output
Apply the above steps to infer a plausible object for this quadruple. Remember, your answer should be grounded in logic and supported by your analysis of the given information. Generate a plausible answer object for this quadruple:
```

Table 8: The *unsupervised+cot* querying prompt template in the experiments of Section 5.4. We replace the colored slot with quadruple before querying the LLMs. Note that we use the same template when conducting *SFT+cot* on LLM.

```
#### [User_INPUT]
## Task
You are provided with a quadruple from a temporal knowledge graph. A temporal quadruple typically consists of four components: subject, verb, time, and object. In this case, the subject, verb, and time are given, but the object is missing.
Your task is to infer the most likely object for this quadruple based on the given subject, verb, and time. Remember, this temporal knowledge graph focuses on international relations and political events, so consider these factors in your reasoning.

## Query Quadruple
Subject: {query_subject}
Verb: {query_relation}
Time: {query_time}
Object: ? (Unknown)

## Output Format
Please provide your answer in the following format:
Object: [Your Final Answer Object]

## Output
Generate a plausible answer object for this quadruple:

#### [LLM_Output]
Object: Germany
```

Table 9: The Instruction-Tuning Data using in SFT.

```
[Query Quadruple]
Subject: Donald Trump
Verb: Sign formal agreement
Time: 2018-10-17
Object: ? (Unknown)

[Response of LLM]
Based on the given input quadruple, it is difficult to infer a specific object for the verb "Sign formal agreement" without more context. However, some possible objects that could be inferred include:
1. A treaty or agreement between two countries or organizations.
2. An executive order or directive signed by the President.
It is also worth noting that the time stamp given suggests that this event occurred on October 17th, 2018. This could provide some additional clues about the context of the agreement being signed, such as the location or topic of the negotiations.
```

Table 10: LLM refuses to answer.