Deja vu: Contrastive Historical Modeling with Prefix-tuning for Temporal Knowledge Graph Reasoning

Miao Peng[1]*, **Ben Liu**[1]*, **Wenjie Xu**[1], **Zihao Jiang**[1], **Jiahui Zhu**[2], **Min Peng**[1]†
[1]School of Computer Science, Wuhan University, China
[2]Xiaomi Inc., China
{pengmiao,liuben123,vingerxu,jiangzihao,pengm}@whu.edu.cn
zhujiahui1@xiaomi.com

Abstract

Temporal Knowledge Graph Reasoning (TKGR) is the task of inferring missing facts for incomplete TKGs in complex scenarios (e.g., transductive and inductive settings), which has been gaining increasing attention. Recently, to mitigate dependence on structured connections in TKGs, text-based methods have been developed to utilize rich linguistic information from entity descriptions. However, suffering from the enormous parameters and inflexibility of pre-trained language models, existing text-based methods struggle to balance the textual knowledge and temporal information with computationally expensive purpose-built training strategies. To tap the potential of text-based models for TKGR in various complex scenarios, we propose **ChapTER**, a **C**ontrastive **h**istoric**a**l modeling framework with **p**refix-tuning for **TE**mporal **R**easoning. ChapTER feeds history-contextualized text into the pseudo-Siamese encoders to strike a textual-temporal balance via contrastive estimation between queries and candidates. By introducing virtual time prefix tokens, it applies a prefix-based tuning method to facilitate the frozen PLM capable for TKGR tasks under different settings. We evaluate ChapTER on four transductive and three few-shot inductive TKGR benchmarks, and experimental results demonstrate that ChapTER achieves superior performance compared to competitive baselines with only 0.17% tuned parameters. We conduct thorough analysis to verify the effectiveness, flexibility and efficiency of ChapTER.

1 Introduction

Knowledge Graphs (KGs) constitute structured representations of knowledge, storing substantial factual information in the form of (*subject*, *prediction*, *object*). KGs have been an essential component of various NLP applications including question answering (Yasunaga et al., 2021), recommendation (Yang et al., 2022), etc. Considering facts inherently evolve in KGs over time, Temporal Knowledge Graphs (TKGs) are constructed to describe the relationship between entities over time in the form of quadruple (*subject*, *prediction*, *object*, *timestamp*). While TKGs are usually incomplete, TKG reasoning (TKGR) aims to predict the missing facts from known ones. In this paper, we focus on the extrapolation task, which requires forecasting future events on TKGs with historical events. For instance, TKGR needs to answer the query (*Olivia Rodrigo*, *Release an album*, *?*, *2023-9-8*) by matching and selecting from all candidate entities based on related historical events.

To address the problem of TKG reasoning, many efforts have been made to capture temporal evolutional information in TKGs. Due to the graph-like features of TKGs, previous methods (Xu et al., 2023b; Zhu et al., 2021; Li et al., 2021) work on designing the temporal-aware encoders referring to known history and mine evolutional patterns from query neighborhoods. Recently, pretrained language models (PLMs) have been showing great abilities to model textual linguistic semantics, and some methods incorporate temporal information of TKGs into PLMs by designing manufactural prompts with fact texts (Xu et al., 2023a), tuning PLMs with a time-specific masking strategy (Chen et al., 2023b), etc. Nevertheless, on the one hand, these manually designed prompts are explicitly based on a priori assumption. On the other hand, training an entire language model on the time-specific masking strategy is computationally expensive. Though PLMs are equipped with strong linguistic inherence from the pre-training stage, they tend to exceedingly focus on the textual semantics, thus struggling to balance time-specific information and textual knowledge in TKGs. Furthermore, given the highly dynamic essence of

*Equal contribution.
†Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1178–1191

TKG, the continuous emergence of new unseen entities usually leads to the need for TKGR to predict entities in a more complex scenario (e.g. few-shot inductive scenario). Thus, in this paper, we focus on the research question: *Can we efficiently integrate temporal history into textual knowledge in a unified PLM-based framework and adapt to TKG reasoning tasks in various complex scenarios?*

To this end, we propose ChapTER, an efficient temporal-aware PLM-based pseudo-Siamese framework adaptable for TKGR in different scenarios. Specifically, ChapTER first verbalizes the input queries and candidates with up-to-date historical contexts, and feeds them into the two-tower model encoders individually to learn history-contextualized embeddings in a decoupled manner. Then contrastive estimation is performed between them to strike a balance of temporal information and textual knowledge in representations. Rather than training an entire model, ChapTER applies a two-tower prefix-based tuning method to enable frozen PLMs capable of performing TKG reasoning tasks under both transductive and few-shot inductive settings. To refrain from the dependency of entity-related prompts in previous method (Chen et al., 2023a), we feed entity-agnostic virtual time prefix prompts into the frozen PLMs, empowering the model to TKGs with unseen entities.

To summarize, our contributions are as follows:

- We explore a unified PLM-based pseudo-Siamese framework that can be efficiently adapted to TKGR tasks in various complex scenarios by utilizing computationally efficient prefix-based tuning.
- ChapTER models the historical contextual information through contrastive learning by enforcing query and candidate with highly correlated history closer and vice versa, which strikes a good balance of temporal information and textual knowledge.
- We evaluate ChapTER on both transductive and few-shot inductive TKGR tasks and experimental results on seven datasets demonstrate ChapTER achieves competitive performance with less than 0.17% tuned parameters.

2 Related Work

Transductive TKG Reasoning. Most existing transductive TKG reasoning methods are embedding-based and some of them extended from previous KG reasoning methods. TTransE (Leblay and Chekol, 2018a) extends the distance-based method TransE (Bordes et al., 2013) by incorporating extra temporal constraints among facts. TNTComplEx (Lacroix et al., 2020) extends ComplEx (Trouillon et al., 2016) by performing 4th-order tensor factorization to learn time-aware representations. Besides, graph-based methods (Jin et al., 2020; Li et al., 2021, 2022a) employ GCNs to capture the structural information via message passing and model temporal correlations from knowledge graph snapshots with historical information. More recently, PLM-based models have been utilized to incorporate external textual semantics for TKG reasoning. PPT (Xu et al., 2023a) converts TKGR task into a masked token prediction task by utilizing PLM with manually designed prompts. ECOLA (Han et al., 2023) learns the contextualized representations by jointly optimizing the TKGR and the masked language modeling objectives.

Inductive TKG Reasoning. TKG reasoning tasks under inductive setting aim to predict new emerging entities in TKGs, indicating that unseen entities in the test set are not contained in the train set. To handle unseen entities, GNN-based methods like GraIL (Teru et al., 2020) and NOODLE (Liu et al., 2023) extract enclosing subgraphs and learn entity-agnostic local structural information. TLogic (Liu et al., 2022) mines entity-independent logic rules to infer unseen entities. SST-BERT (Chen et al., 2023b) conducts inductive relation prediction by applying a time masking MLM task to pre-train BERT with structured sentences. FILT (Ding et al., 2022) adopts a meta-learning-based model with entity concepts to handle unseen entities in TKGs.

Prefix Tuning. Prompts are manually designed textual templates to query a language model, and they are beneficial to help language models solve different tasks with all parameters frozen. To alleviate the suboptimal performance caused by discrete prompting, continuous prompts with trainable embeddings are added to the embeddings of input sequence (Liu et al., 2021b; Lester et al., 2021), which have been shown to achieve competitive performance across various NLP tasks. Li and Liang (2021) adds trainable prefix vectors to each transformer layer within frozed Seq2Seq PLMs, aiming to efficiently adapt PLMs to natural generation tasks. Chen et al. (2023a) introduces conditional soft prompts to sufficiently incorporate textual se-

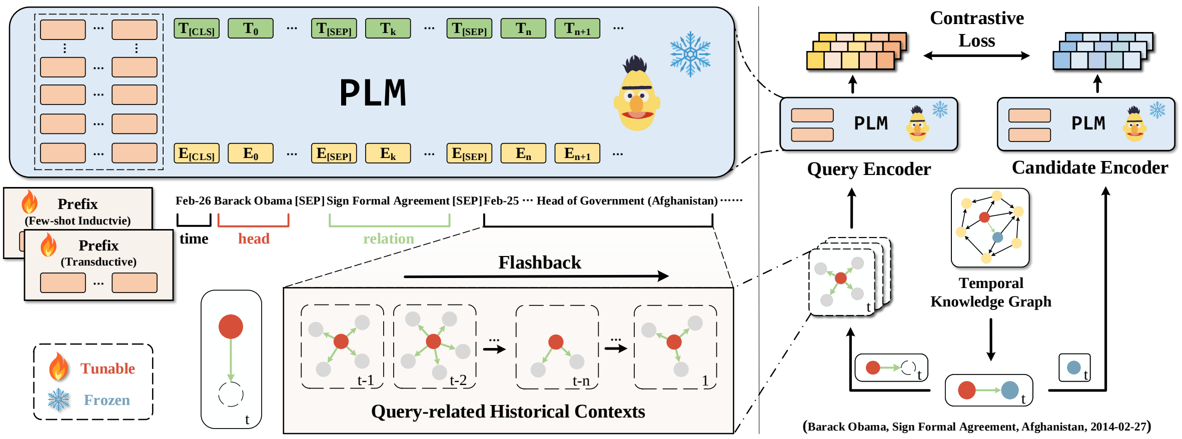


Figure 1: Overall illustration of the ChapTER model: 1) an example of ChapTER for the TKG input query (*Barack Obama, Sign Formal Agreement, ?, 2014-02-07*) and corresponding candidate *Afganistan* (right); 2) a detailed sketch about the structure and verbalized input of Query Encoder (left).

mantics into structural information for KGC tasks. Jin et al. (2023) integrates local structure information into transformer layer text encoding via virtual node tokens.

3 Method

In this section, we first give out the preliminaries and formulation of *Temporal Knowledge Graph Reasoning* in Sec.3.1. Then we introduce the detail model framework from Sec.3.2 to Sec.3.5.

3.1 Preliminaries

Temporal Knowledge Graph (TKG). TKG is a directed graph with a collection of fact quadruples. Let $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$ be a TKG instance, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$ represent the set of entities, relations and timestamps respectively. $\mathcal{F}$ denotes the set of quadruples (s, p, o, t), in which $s \in \mathcal{E}$ is a subject (head) entity, $o \in \mathcal{E}$ is an object (tail) entity, and $r \in \mathcal{R}$ is the predicate (relation) appearing at time t between s and o. Under this definition, a TKG can be represented as a sequence of KGs $\{\mathcal{E}, \mathcal{R}, \mathcal{F}_t\}$, where $\mathcal{F}_t$ is the set of facts that occurred at time t.
Transductive TKG Reasoning. The TKG reasoning task under transductive setting aims to answer the queries including $(s, p, ?, t_q)$ and $(?, p, o, t_q)$. Following the extrapolation setting (Zhu et al., 2021), training, validation and test sets are KGs from timestamps T_0 to T_1, T_1 to T_2, T_2 to T_3 ($T_0 < T_1 < T_2 < T_3$).
Few-shot Inductive TKG Reasoning. Under the inductive setting of TKG reasoning, given an observed background TKG $\mathcal{G}_B \subseteq \mathcal{E}_B \times \mathcal{R} \times \mathcal{E}_B \times \mathcal{F}$, unseen entity e' is a fact from the set $\mathcal{E}'$, where $\mathcal{E}_B \cap \mathcal{E}' = \emptyset$. Hypothesizing that there are K observed associated quadruple facts for each unseen entity e', denoting as (e', p, e^*, t) or (e^*, p, e', t), where $e^* \in \mathcal{E}_B \cup \mathcal{E}'$. The goal of inductive few-shot TKGC reasoning is to answer the queries like $(e', p, ?, t_q)$ or $(?, p, e', t_q)$ from unobserved quadruples with unseen entities.

3.2 The ChapTER Model Framework

By converting the TKG reasoning problem into a query-candidate matching problem, the goal of ChapTER Model is to model the historical information and balance them with the textual semantics appropriately via a contrastive manner.

As illustrated in Figure 1, ChapTER is a pseudo-Siamese network consisting of two encoders: the query encoder $\mathcal{M}_q$ and the candidate encoder $\mathcal{M}_k$. To encode the textual information from entities and relations, we adopt the transformer-based PLM as our encoder, e.g. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). $\mathcal{M}_q$ and $\mathcal{M}_k$ are initialized with the same weight but tuned with prefix prompts separately. Given the query $q = (s, p, ?, t_q)$ under the tail prediction setting, $\mathcal{M}_q$ is the query encoder that aims to obtain the time-conditional entity-relation embedding $\boldsymbol{h}_q$ containing textual information of (s, p) pair with historical contexts constrained on time t. Similarly, the candidate encoder $\mathcal{M}_k$ encodes the textual embedding $\boldsymbol{h}_k$ of candidate entity o. We take the mean pooling of the last-layer hidden state from $\mathcal{M}_q$ and

$\mathcal{M}_k$ as the embeddings of $\boldsymbol{h}_q$ and $\boldsymbol{h}_k$:

$$\boldsymbol{h}_q = \mathcal{M}_q(\mathcal{P}(q)),\ \boldsymbol{h}_k = \mathcal{M}_k(\mathcal{P}(k)), \quad (1)$$

where $\mathcal{P}(q)$ and $\mathcal{P}(k)$ denote the input text of query and candidate separately.

With the obtained embeddings $\boldsymbol{h}_q$ and $\boldsymbol{h}_k$, the score of a quadruple (s, p, o, t) can be regarded as the cosine similarity between $\boldsymbol{h}_q$ and $\boldsymbol{h}_k$ by simply performing a dot-product of these two embeddings:

$$f(s,p,o,t) = \cos(\boldsymbol{h}_q, \boldsymbol{h}_k) = \frac{\boldsymbol{h}_q \cdot \boldsymbol{h}_k}{\|\boldsymbol{h}_q\| \, \|\boldsymbol{h}_k\|}. \quad (2)$$

Hence, for the query q, we compute the cosine similarity between query embedding $\boldsymbol{h}_q$ and all other candidate entity embeddings $\boldsymbol{h}_k$, and take the one with the highest score as the final prediction:

$$\arg\max_{k_i} \cos(\boldsymbol{h}_q, \boldsymbol{h}_{k_i}),\ k_i = o_i \in \mathcal{E}. \quad (3)$$

3.3 Text Representation & Verbalization

Fact Description. To represent the entity or relation in a quadruple, the most representative information is the name text, e.g. "*Zalmai Rassoul*" for entity and "*Make statement*" for relation in dataset ICEWS (Integrated Crisis Early Warning System) (Boschee et al., 2015). Despite this, the name texts often turn out to be short and highly overlapped, which may refer to multiple entities in the training corpus of PLM models and lead to the problem of ambiguity. To avoid this problem, we enrich the expression of semantic information by introducing the description text. More concretely, we formulate entity descriptions of ICEWS by including the hierarchical text from *Country* field and *Sector* field. For example, the description of entity "*Virtue Party*" is "*Turkey, Sunni International Religious*". Based on this, we concatenate the name and description text together as the complete entity description D_s. As for relation, we directly use its name text D_r.

Verbalization with Historical Context. Given a candidate quadruple fact (s, p, o, t) under the tail entity prediction task, we divide it into two parts: the time-conditional entity-relation query $q = (s, p, ?, t_q)$ and the candidate entity $k = o$. For the query q, we hypothesize that the historical information remains in the neighbor pairs of entity-relation-related context, which contains both s and p of query q. Specifically, we define the set of historical quadruples as follows:

$$\mathcal{H}(s,p,t) = \big\{(s,p,\tilde{o},t') \mid (s,p,\tilde{o},t') \in \mathcal{F}, \\ \tilde{o} \neq o, t' \leq t\big\}. \quad (4)$$

With historical context of the quadruple fact, we individually represent the query q and candidate k into two different prompt formats. For timestamp text t in format of "yyyy-mm-dd", we replace its month number with corresponding lexical text L_t. Formally, we have the input $\mathcal{P}(q)$ of query q as follows:

$$\langle cls \rangle\ L_t \mid D_s \mid D_r\ \langle sep \rangle\ \mathcal{H}(s,p,t)\ \langle sep \rangle$$

where D_s and D_r denotes the verbalized text of entity s and relation r. Likewise, we have the input $\mathcal{P}(k)$ of candidate k as:

$$\langle cls \rangle\ D_s\ \langle sep \rangle$$

3.4 Text encoding with virtual time prefix

The goal of ChapTER is to model both historical information and textual semantics from TKGs in various complex scenarios. Instead of designing a time-specific masking strategy to pre-train a model from scratch, we introduce the virtual time prefix tokens to each Transformer layer within PLM, aiming to inject both historical and semantic information into the two-tower transformer-based model encoding procedure. We apply the prefix-based tuning methods to equip our model with capabilities to handle both transductive and few-shot inductive setting tasks. Previous works (Li and Liang, 2021; Liu et al., 2021a) have shown the effectiveness of prefix-tuning methods in facilitating models the ability to different tasks, while achieving comparable performance with only a few parameters tuned.

We employ vector $\boldsymbol{h}$ to uniformly represents $\boldsymbol{h}_q$ in $\mathcal{M}_q$ and $\boldsymbol{h}_k$ in $\mathcal{M}_k$. Denoting $\boldsymbol{h}^{(j)} \in \mathcal{R}^{n \times d}$ as the output embeddings of all tokens in input text after j-th ($i \geq 1$) transformer layer, we concatenate the virtual time prefix $\boldsymbol{p}$ with m-length token embeddings to the text token embeddings in each transformer layer as follows:

$$\widetilde{\boldsymbol{h}}^{(j)} = \boldsymbol{p}^{(j)} \parallel \boldsymbol{h}^{(j)}, 0 \leq j \leq L, \quad (5)$$

where $\boldsymbol{p}^{(j)}$ indicates the virtual prefix token embeddings of j^{th} layer and $\widetilde{\boldsymbol{h}}$ is the concatenated input token embeddings. Concretely, the i^{th} input token of the j^{th} layer is defined as:

$$\widetilde{\boldsymbol{h}}_i^{(j)} = \begin{cases} \boldsymbol{p}_i^{(j)}, & 0 \leq i < m \\ \boldsymbol{e}_i^{(j)}, & (i \geq m) \wedge (j = 0) \\ \text{FFN}(\widetilde{\boldsymbol{h}}^{(j-1)})_i, & (i \geq m) \wedge (j \geq 1) \end{cases} \tag{6}$$

During the training procedure, the weight parameters of PLM models $\mathcal{M}_q$ and $\mathcal{M}_k$ are frozen and only weight parameters in prefix prompts are updated in parallel, in which we apply the multi-head attention mechanism as follows:

$$\begin{aligned} \text{MHA}(\boldsymbol{h}^{(j)}, \widetilde{\boldsymbol{h}}^{(j)}) &= \big\Vert_{i=1}^{k} \text{head}_i(\boldsymbol{h}_i^{(j)}, \widetilde{\boldsymbol{h}}_i^{(j)}) \\ &= \big\Vert_{i=1}^{k} \text{softmax}(\mathbf{Q}^{(j)}, \widetilde{\mathbf{K}}^{(j)\top})\widetilde{\mathbf{V}}^{(j)}, \end{aligned} \tag{7}$$

$$\begin{gathered} \mathbf{Q}^{(j)} = \mathbf{W}_Q^{(j)} \boldsymbol{h}^{(j)}, \\ \widetilde{\mathbf{K}}^{(j)} = \mathbf{W}_K^{(j)} \widetilde{\boldsymbol{h}}^{(j)},\ \widetilde{\mathbf{V}}^{(j)} = \mathbf{W}_V^{(j)} \widetilde{\boldsymbol{h}}^{(j)}. \end{gathered} \tag{8}$$

We keep the query ($\mathbf{Q}$) vector still but enhance the key ($\mathbf{K}$) and value ($\mathbf{V}$) vectors with prefix embeddings. By performing the asymmetric multi-head attention in each layer, prefix vectors can efficiently capture specific data characteristics in different datasets with only a few parameters tuned.

3.5 Training and Inference

With representations of query $\boldsymbol{h_q}$ and candidate $\boldsymbol{h_k}$, in the training procedure, we apply the InfoNCE loss (van den Oord et al., 2018; Peng et al., 2022) to perform contrast estimation as follows:

$$\mathcal{L}_{cl} = -\log \frac{e^{(cos(\boldsymbol{h_q}, \boldsymbol{h_k}) - \gamma)/\tau}}{e^{(cos(\boldsymbol{h_q}, \boldsymbol{h_k}) - \gamma)/\tau} + \sum_{i \in \mathcal{N}_{neg}} e^{cos(\boldsymbol{h_q}, \boldsymbol{h}'_i)/\tau}}, \tag{9}$$

where τ is a learnable temperature parameter and γ ($\gamma > 0$) is the additive margin that encourages the model to score higher for correct quadruples.

$\mathcal{N}_{neg}$ represents the set of negative samples during training. Instead of randomly corrupting s or p of existing quadruples, we formulate $\mathcal{N}_{neg}$ with three types of negative samples:

$$\mathcal{N}_{neg} = \{o' \mid o' \in \mathcal{N}_{in} \cup \mathcal{N}_{pre} \cup \mathcal{N}_{self}\}. \tag{10}$$

Specifically, $\mathcal{N}_{in}$ represents the set of in-batch negatives, meaning that entities within the same batch can be taken as the negative sample of each other. As for pre-batch negatives $\mathcal{N}_{pre}$, we employ a dynamic queue to store entities from recent previous k batches. Besides, we take head entity s from tail prediction query $(s, p, ?, t_q)$ as hard self-negative $\mathcal{N}_{self}$ to diminish false predictions due to the high text overlap between query and head entity.

For inference, ChapTER first obtains the embeddings of query $(s, p, ?, t_q)$ and all candidates via $\mathcal{M}_q$ and $\mathcal{M}_k$ separately, then computes the entity ranking by the dot-product scores between them.

4 Experimental Setup

Datasets. We evaluate ChapTER on TKGR task in both transductive and few-shot inductive settings. For transductive TKGR, we use four widely-used event-based TKG datasets: ICEWS14, ICEWS18, ICEWS05-15 (Han et al., 2021) and ICEWS14* (Li et al., 2022b). For few-shot inductive TKGR, we use three TKG few-shot OOG benchmarks proposed in Ding et al. (2022): ICEWS14-OOG, ICEWS18-OOG and ICEWS0515-OOG. For textual descriptions, existing ICEWS datasets do not provide entity description texts, so we create them by combining corresponding *country* and *sector* entries for each entity. Detailed dataset statistics are shown in Appendix A.1.

Implementation Details. All experiments are carried out on 24G RTX 3090. We adpot AdamW optimizer with linear learning rate decay to train ChapTER. The query encoder and candidate encoder are initialized with parameters of *bert-based-uncased*. We truncate the description token length up to 50 for entities. The learnable temperature τ is initialized to 0.05 and the additive margin is set to 0.02. We formulate pre-batch negatives N_{pre} from previous 2 batches. For the settings of all baselines, we adopt their default configurations. Most of the transductive TKGR results are taken from Han et al. (2021) and few-shot inductive TKGR results are taken from Ding et al. (2022). For fairness of comparison, we reimplemented SimKGC, KGT5, KGT5-context based on their open source codes to adequately incorporate temporal information. We report the metrics MRR (mean reciprocal rank) and Hits@N (proportion of correct entity rank) to evaluate the performance of ChapTER. We calculate the model results under the time-aware filtered setting (Li et al., 2022b). More detailed implementation settings can be found in Appendix A.3, A.4 and A.5. Codes are avaliable at this website[1].

[1] https://github.com/GKNL/ChapTER

Model	ICEWS14			ICEWS18			ICEWS05-15			ICEWS14*		
	MRR	H@3	H@10	MRR	H@3	H@10	MRR	H@3	H@10	MRR	H@3	H@10
Graph-Based Methods												
DistMult (Yang et al., 2015)	.162	.179	.253	.102	.103	.213	.287	.322	.475	.154	.172	.239
ComplEx (Trouillon et al., 2016)	.213	.231	.352	.210	.235	.399	.317	.357	.520	.325	.361	.507
RotatE (Sun et al., 2019)	.209	.239	.440	.128	.149	.319	.247	.290	.482	.213	.244	.448
TTransE (Leblay and Chekol, 2018a)	.134	.173	.346	.083	.086	.219	.157	.197	.380	.137	.177	.357
TA-DistMult (García-Durán et al., 2018)	.265	.302	.454	.168	.184	.336	.265	.302	.454	.258	.297	.430
DE-SimplE (Goel et al., 2020)	.327	.357	.491	.193	.219	.348	.350	.390	.528	.334	.372	.498
TNTComplEx (Lacroix et al., 2020)	.321	.360	.491	.212	.240	.369	.275	.308	.429	.340	.385	.509
CyGNet (Zhu et al., 2021)	.327	.363	.507	.249	.283	.426	.350	.391	.529	.351	.390	.536
PLM-Based Methods												
SimKGC (Wang et al., 2022)	.267	.289	.413	.210	.235	.349	.309	.337	.472	.264	.287	.409
KGT5 (Saxena et al., 2022)	.261	.297	.453	.221	.250	.396	.264	.295	.411	.217	.238	.351
KGT5-context (Kochsiek et al., 2023)	.280	.333	.478	.228	.267	.411	.304	.362	.489	.323	.355	.508
ChapTER	**.332**	**.370**	**.515**	**.244**	**.276**	**.412**	**.331**	**.369**	**.525**	**.338**	**.380**	**.527**

Table 1: Transductive TKG reasoning performance (with time-aware metrics) on ICEWS14, ICEWS18, ICEWS05-15 and ICEWS14*. The best PLM-based method results are in **bold** and the second best results are underlined. For fair comparison, we add corresponding timestamps of quadruples into the input text for PLM-based baselines, to equip them with the capacities of modeling time information. More results on WIKI and YAGO datasets can be found in Appendix B.

5 Experimental Results

In this section, we first compare ChapTER against other competitive baselines in both transductive and few-shot inductive TKG reasoning tasks in Sec 5.1. Then we conduct ablation study in Sec 5.2 to evaluate the effectiveness of each component in ChapTER. After that, we further analyze the efficiency and flexibility of ChapTER in Sec 5.3.

5.1 Main Results

We compared our proposed ChapTER with various competitive baselines, and the main results of transductive and few-shot inductive TKG reasoning summarized in Table 1 and Table 2, respectively.

Results on Transductive TKGR. On transductive TKGR benchmarks, we compare ChapTER with both graph-based and PLM-based models. Results on four datasets show that ChapTER achieves state-of-the-art or competitive performance against baselines. Specifically, on ICEWS14 dataset, ChapTER outperforms all PLM-based methods by a substantial margin and achieves 18.5% (from .280 to .332) relative MRR improvement. It is worth noting that ChapTER achieves better performance with only a few prefix parameters tuned compared to fully trained PLM-based baselines, which verifies the effectiveness of prefix-tuning in TKGR tasks.

Compared with graph-based methods, ChapTER consistently outperforms previous baselines on ICEWS14 (MRR .332 v.s. .327), and the competitive results demonstrate ChapTER holds superiority of modeling representations in future timestamps through historical contexts. We also find that ChapTER maintains modest results on ICEWS18 and ICEWS05-15. It is worth noting that events involved in these datasets are more dense and frequent with more entities, indicating that more events are happening in the same timestamp. Since CyGNet is designed to capture the facts recurrence in the appeared history, it is good at predicting events with repetitive history yet inferior in absorbing TKG texts. This explains why ChapTER marginally lags behind CyGNet on these datasets, because redundant historical events text are truncated due to the limitation of input length in PLMs.

Results on Few-shot Inductive TKGR. As shown in Table 1, we verify ChapTER's TKG reasoning performance in a more complex few-shot inductive scenario on ICEWS14-OOG, ICEWS18-OOG and ICEWS0515-OOG datasets, considering both 1-shot and 3-shot settings. It can be seen that ChapTER substantially outperforms existing few-shot inductive TKGR methods. Concretely, ChapTER achieves striking improvement in hit@10 on ICEWS14-OOG (from .410 to .750 in 1-shot, from .475 to .761 in 3-shot), though being slightly worse on Hit@3 (3-shot) than FILT. We also report the zero-shot performance of ChapTER on these three datasets, and we can observe that ChapTER consistently outperforms all baselines, though slightly lags behind on few-shot performances. The overall remarkable performance verifies that ChapTER can

Model	ICEWS14-OOG						ICEWS18-OOG						ICEWS0515-OOG					
	MRR		H@3		H@10		MRR		H@3		H@10		MRR		H@3		H@10	
	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S	1-S	3-S
Inductive TKGR Methods																		
ComplEx (Trouillon et al., 2016)	.048	.046	.045	.046	.099	.089	.039	.044	.048	.042	.085	.093	.077	.076	.074	.071	.129	.120
BiQUE (Guo and Kok, 2021)	.039	.035	.041	.030	.073	.066	.029	.032	.033	.037	.064	.073	.075	.083	.072	.077	.130	.144
TNTComplEx (Lacroix et al., 2020)	.043	.044	.033	.042	.102	.096	.046	.048	.043	.044	.087	.082	.034	.037	.031	.036	.060	.071
TeLM (Xu et al., 2021)	.032	.035	.021	.023	.063	.077	.049	.019	.045	.013	.084	.054	.080	.072	.077	.072	.138	.151
TeRo (Xu et al., 2020)	.009	.010	.005	.002	.015	.020	.007	.006	.006	.003	.013	.006	.012	.023	.008	.017	.024	.040
MEAN (Hamaguchi et al., 2017)	.035	.144	.032	.145	.082	.339	.016	.101	.012	.114	.043	.283	.019	.148	.017	.175	.052	.384
LAN (Wang et al., 2019)	.168	.199	.199	.255	.421	.500	.077	.127	.067	.165	.199	.344	.171	.182	.180	.191	.367	.467
GEN (Baek et al., 2020)	.231	.234	.250	.284	.378	.389	.171	.216	.189	.252	.289	.351	.268	.322	.308	.362	.413	.507
FILT (Ding et al., 2022)	.278	.321	.305	.357	.410	.475	.191	.266	.209	**.298**	.316	.417	.273	**.370**	.303	**.391**	.405	.516
PLM-Based Methods																		
SimKGC (Wang et al., 2022)	.346	.363	.399	.426	.705	.721	.243	.252	.280	.295	**.549**	.555	.312	.318	**.375**	.382	.629	.637
ChapTER	**.364**	**.379**	**.428**	**.446**	**.750**	**.761**	**.257**	**.266**	**.284**	.296	.547	**.558**	**.319**	.323	.368	.373	**.644**	**.648**
ChapTER - Zero Shot	.361		.419		.752		.257		.284		.544		.315		.368		.638	

Table 2: Few-shot inductive TKG reasoning performance on ICEWS14-OOG, ICEWS18-OOG and ICEWS0515-OOG. The best results are in **bold** and the second best results are underlined.

No.	Model	ICEWS14		ICEWS14-OOG	
		MRR	H@10	MRR	H@10
1	ChapTER	**.332**	**.515**	**.361**	.752
2	*w/o* timestamp	.321	.505	.358	.739
3	*w/o* description	.319	.497	.308	.651
4	*w/o* historical contexts	.324	.484	.350	.706
5	*w/o* pre-batch neg	.326	.509	.350	.726
6	*w/o* self neg	.329	.510	.358	**.755**
7	*w/o* pre-batch & self neg	.326	.507	.343	.734

Table 3: Ablation study of components in ChapTER on ICEWS14 and ICEWS14-OOG (zero-shot). Lines 2-4 report variants on input prompt composition, lines 5-7 report variants on negatives combination.

transfer knowledge from known training entities to unseen ones, with prior knowledge and encoded historical information in PLMs.

5.2 Ablation Studies

To further analyze how each component of ChapTER contributes to the final performance, we conduct ablation studies on ICEWS14 and ICEWS14-OOG, and complete results are reported in Table 3. More ablation study results on training strategies and prompt length can be found in Appendix C.

Input Text Composition. To verify the effectiveness of text verbalization approach mentioned in Sec. 3.3, we consider three variants by selectively removing the timestamp text, description text and historical contexts separately. In Table 3, lines 2-4 show the performance of ChapTER with different input compositions. Compared to ChapTER, the performance of each ablated variant exhibits marginal decreases. Intuitively, removing description texts produces the largest performance drop (MRR drops 3.9% and 14.7% separately), since PLM-based models fundamentally rely on text quality. With more informative entity descriptions, the performance of ChapTER can be further improved. Moreover, lines 3-4 support the importance of temporal historical information for ChapTER. We argue this phenomenon for two reasons: 1) Though timestamps contain essential temporal information, they tend to be terse and drowned out by verbose text; 2) In contrast, historical contexts contribute more substantially to temporal modeling, as they introduce sequenced, up-to-date histories that provide abundant background for queries. Thus, ChapTER models can capture more accurate temporal information from rich historical contexts.

Negative Sample Combination. Table 3 Lines 5-7 show the performance of ChapTER with different training negatives (in-batch $\mathcal{N}_{in}$, pre-batch $\mathcal{N}_{pre}$ and self negatives $\mathcal{N}_{self}$) on ICEWS14 and ICEWS14-OOG. Three ablated variants were evaluated by separately removing $\mathcal{N}_{pre}$, $\mathcal{N}_{self}$, or both from ChapTER. We observe that removing both $\mathcal{N}_{pre}$ and $\mathcal{N}_{self}$ yields worse empirical results than removing them separately. It's worth noting that since self-negatives diminish the rely of ChapTER on naive text match, they tend to improve Hits@1 but hurt Hits@10 (e.g. Hits@10 from .755 to .752 on ICEWS14-OOG). Furthermore, we investigate the impact of in-batch negative sample numbers during model training, as shown in Figure 2. By increasing the number of negative samples, there is a steady improvement from .192 to .322, but it only

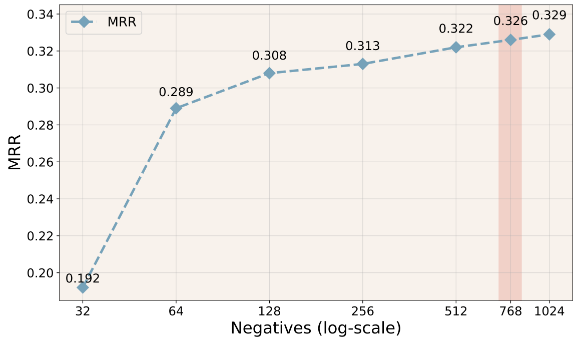


Figure 2: MRR results on ICEWS14 with in-batch negatives number changing in ChapTER.

Model	PLM	Total	Trainable	T_{train}/ep
SimKGC	Bert-base Bert-large	218.9M 670.3M	218.9M 670.3M	3.2min 20.0min
KGT5	T5-small T5-base	60.5M 223M	60.5M 223M	27.6min 42.6min
KGT5-context	T5-small T5-base	60.5M 223M	60.5M 223M	38.5min 55.5min
ChapTER	Bert-base Bert-large	219.3M 671.3M	0.37M 0.99M	3.3min 14.0min

Table 4: Model Efficiency of ChapTER on ICEWS14 comparing to other PLM-based methods. *Total* and *Trainable* indicates the total and trainable parameters.

obtains slight change when the number is larger than 768 (red bar). In summary, each kind of negative contributes to the best results of ChapTER.

5.3 Discussion

In this section, we conduct further analysis on model efficiency, the ability to capture temporal information, the impact of different tuning strategies and the impact of different PLM models.

Q1: How efficient ChapTER is compared to other PLM-based models? Table 4 summarizes the model efficiency of ChapTER compared to other PLM-based methods. By taking advantage of the efficient tuning strategy, ChapTER achieves superior performance with minimal parameters tuned and reduced training time. Compared with SimKGC, ChapTER is 1.4x time faster in training with only 0.15% parameters tuned (0.99M v.s. 760.3M). Considering recently proposed sequence-to-sequence KGR models, ChapTER outperforms KGT5 with 0.6% parameters trained and 12x faster training time, this is because KGT5 needs to train a T5 model from scratch with task-specific input prompts. Besides, during inference, KGT5 is computationally expensive (0.83min v.s. 95.23min) due to a huge decoding search space. This suggests that ChapTER is more efficient in time and computation while achieving superior performance.

Model	ICEWS14		ICEWS14-OOG	
	MRR	H@10	MRR	H@10
ChapTER	**.332**	**.515**	**.361**	**.752**
Timestamp Text				
w/o timestamp	.321	.505	.358	.739
random timestamp	.319	.497	.355	.732
Historical Sequence				
history descending order	.332	.515	.361	.752
history ascending order	.322	.498	.344	.721
history random order	.328	.504	.349	.743
Form of Context				
pairs	.332	.515	.361	.752
entities	.324	.502	.359	.734

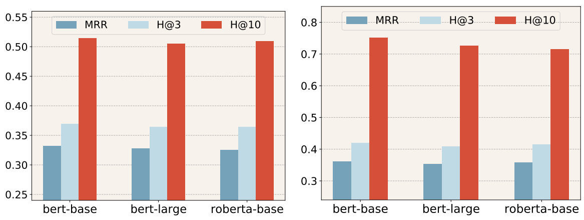
Table 5: Performance of different historical modeling approaches on ICEWS14 and ICEWS14-OOG datasets.

Q2: How does ChapTER use the temporal history information of events? We further investigate how ChapTER actually utilizes the historical context information. As shown in Table 5, we analyze the impact of history modeling in three aspects: "Timestamp Text", "Historical Sequence Order" and "Form of Context". We can observe that removing timestamps or using random timestamps in text input both lead to a performance drop. As for historical sequence, we find that model with history in a descending order performs better than those with ascending or random order. It evidences that recent historical events are more decisive to future forecasting. Besides, we formalize the historical contexts in two ways: *Entity* (e.g., all historical entities that are related to query) and *Pair* (e.g., a list of complete historical quadruples). Results show that concatenate contexts by pairs achieve a higher performance than entities. We believe this is because paired contexts provide more concrete and sequenced event history. In summary, ChapTER is capable of modeling temporal information from recent and complete historical contexts.

Q3: How do different tuning strategies affect ChapTER's performance? As mentioned in Sec.3.4, ChapTER is tuned with virtual prefix prompts with the PLM parameters frozen. To further discuss the impact of different prefix tuning methods, we compare two widely used approaches: Prefix-tuning (Li and Liang, 2021) and P-tuning V2 (Liu et al., 2021a). As summarized in Table 6, we can observe ChapTER (P-tuning V2 with MLP

Method	Re-param	N_{param}	T_{train}/ep	MRR	H@3	H@10
ChapTER - Prefix Tuning	MLP	47.3M	4.2min	.321	.358	.503
ChapTER - P-tuning v2	MLP	19.7M	3.6min	**.332**	**.370**	**.515**
ChapTER - P-tuning v2	Embedding	0.37M	3.3min	.308	.345	.491

Table 6: ICEWS14 results of ChapTER with different prefix tuning methods. *Re-param* denotes the reparameterization encoder and *Num-param* denotes the corresponding trainable parameter numbers (on Bert-base-uncased).



(a) ICEWS14 (b) ICEWS14-OOG

Figure 3: Comparison of ChapTER with different PLM models on ICEWS14 and ICEWS14-OOG datasets.

reparameterization encoder) achieves better performance than the one with prefix tuning (MRR .332 v.s. .321). We also evaluate the impact of reparameterization module on ChapTER with P-tuning v2. The result show that more parameters in MLP bring a marginally improvement on performance, but its effect is inconsistent across datasets and task settings.

Q4: How do PLM models affect ChapTER's performance? Figure 3 compares the model performance of ChapTER with different PLM models. We can observe that ChapTER with three PLM models all achieve close high performance on two datasets, and the utilization of Bert-base yields a marginally better result. This result suggests that our ChapTER is is robust to different PLMs with varying parameters and agnostic to PLM size. Beyond this, it is possible to improve the performance of ChapTER with some PLMs that have longer input contexts (e.g., longformer), and we leave such extensions for future studies.

6 Conclusion and Future Work

In this paper, we propose ChapTER, a PLM-based pseudo-Siamese framework that models balanced textual knowledge and historical information. With the introduced time prefix tokens, ChapTER is capable for TKG reasoning tasks in various complex scenarios through prefix-based tuning. Experimental results on two TKGR tasks demonstrate the superiority of ChapTER compared to competitive baselines. Thorough analysis shows the efficiency and flexibility of ChapTER. In the future, we would like to explore 1) bridging the gap of two towers with shared time prefix tuning; 2) extending our method to Seq2Seq PLMs to model temporal knowledge in a generative manner.

Limitations

ChapTER is able to balance the textual knowledge and temporal information for the TKGR tasks in various scenarios. However, 1) ChapTER is based on PLMs and it relies on unstructured texts like entity names and descriptions. Thus the performance of ChapTER can be affected due to the quality of texts, and it could be further improved on datasets with more informative texts. Compared to ICEWS datasets, in which we manually construct description texts by concatenating the *Country* and *Sector* fields, datasets like Wikidata containing more informative text descriptions may result in better. 2) Due to the essence of virtual tokens in prefix tuning, which contain few parameters to be tuned compared to frozen PLMs, it may cause a collapse on tiny datasets with sparse quadruples and entities. Besides, an appropriate choice of prefix length and learning rate is crucial. We plan to work on these issues in the future work.

Acknowledgements

We would like to thank all the anonymous reviewers and area chairs for their comments. This research is supported by National Science and Technology Major Project (No.2021ZD0113304), National Natural Science Foundation of China (U23A20316), Key R&D Project of Hubei Province (2021BAA029), General Program of Natural Science Foundation of China (NSFC) (Grant No.62072346), and founded by Joint&Laboratory on Credit Technology.

References

Jinheon Baek, Dong Bok Lee, and Sung Ju Hwang. 2020. Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Process-*

ing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

E Boschee, J Lautenschlager, S O'Brien, S Shellman, J Starz, and M Ward. 2015. Integrated crisis early warning system (icews) coded event data. *URL: https://dataverse. harvard. edu/dataverse/icews*.

Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023a. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11489–11503.

Zhongwu Chen, Chengjin Xu, Fenglong Su, Zhen Huang, and Yong Dou. 2023b. Incorporating structured sentences with time-enhanced BERT for fully-inductive temporal relation prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 889–899.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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. In *4th Conference on Automated Knowledge Base Construction*.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821.

Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3988–3995.

Jia Guo and Stanley Kok. 2021. Bique: Biquaternionic embeddings of knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8338–8351.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1802–1808.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Zhen Han, Ruotong Liao, Jindong Gu, Yao Zhang, Zifeng Ding, Yujia Gu, Heinz Koeppl, Hinrich Schütze, and Volker Tresp. 2023. ECOLA: enhancing temporal knowledge embeddings with contextualized language representations. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5433–5447.

Bowen Jin, Yu Zhang, Yu Meng, and Jiawei Han. 2023. Edgeformers: Graph-empowered transformers for representation learning on textual-edge networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6669–6683. Association for Computational Linguistics.

Adrian Kochsiek, Apoorv Saxena, Inderjeet Nair, and Rainer Gemulla. 2023. Friendly neighbors: Contextualized sequence-to-sequence link prediction. In *Proceedings of the 8th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2023, Toronto, Canada, July 13, 2023*, pages 131–138.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Julien Leblay and Melisachew Wudage Chekol. 2018a. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1771–1776.

Julien Leblay and Melisachew Wudage Chekol. 2018b. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1771–1776. ACM.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597.

Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022a. Complex evolutional pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 290–296.

Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022b. Hismatch: Historical structure matching based temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7328–7338.

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 '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 408–417.

Ben Liu, Miao Peng, Wenjie Xu, and Min Peng. 2023. Neighboring relation enhanced inductive knowledge graph link prediction via meta-learning. *World Wide Web (WWW)*, 26(5):2909–2930.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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 *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 4120–4127.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org.

Miao Peng, Ben Liu, Qianqian Xie, Wenjie Xu, Hua Wang, and Min Peng. 2022. Smile: Schema-augmented multi-level contrastive learning for knowledge graph link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4165–4177. Association for Computational Linguistics.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2814–2828.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2071–2080.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4281–4294.

Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. 2019. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7152–7159. AAAI Press.

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 2021, Online, June 6-11, 2021*, pages 2569–2578.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1583–1593.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023a. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7790–7803.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023b. Temporal knowledge graph reasoning with historical contrastive learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 4765–4773.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1434–1443.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546.

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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4732–4740.

A Experimental Details

A.1 Dataset

We use the transductive TKGR datasets ICEWS14, ICEWS18, ICEWS05-15, YAGO from Han et al. (2021), WIKI from Leblay and Chekol (2018b) and ICEWS14* from Li et al. (2021), which contain political facts of Integrated Crisis Early Warning System (Boschee et al., 2015). We take few-shot inductive datasets ICEWS14-OOG, ICEWS18-OOG and ICEWS0515-OOG from Ding et al. (2022). Following the original data split, we summarize the statistics of these datasets in Table 7 and Table 8. For the absent description texts, we find the texts of *country* and *sector* entries from origin data source[2] and combining them together to construct entity descriptions.

A.2 Baselines

We compare ChapTER with several competitive state-of-the-art baselines in transductive and few-shot inductive TKG reasoning settings. For transductive TKG reasoning, we include 1) traditional KG reasoning methods, i.e. DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) and RotatE (Sun et al., 2019); 2) TKG Reasoning methods, i.e. TTransE (Leblay and Chekol, 2018a), TA-DistMult (García-Durán et al., 2018), TA-TransE (García-Durán et al., 2018), DE-SimplE (Goel et al., 2020), TNTComplEx (Lacroix et al., 2020) and CyGNet (Zhu et al., 2021); 3) PLM-based methods, i.e. SimKGC (Wang et al.,

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

Dataset	$\lvert\mathcal{F}_{train}\rvert$	$\lvert\mathcal{F}_{valid}\rvert$	$\lvert\mathcal{F}_{test}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Time Snapshot	Time Granularity
ICEWS14	63,685	13,823	13,222	7,128	230	365	1 day
ICEWS18	373,018	45,995	49,545	23,033	256	304	1 day
ICEWS05-15	322,958	69,224	69,147	10,488	251	4,017	1 day
ICEWS14*	74,845	8,514	7,371	7,128	230	365	1 day
WIKI	539,286	67,538	63,110	12,554	24	232	1 year
YAGO	51,205	10,973	10,973	100,38	10	194	1 year

Table 7: Statistics of datasets for transductive TKG reasoning. $|\mathcal{F}_{train}|$, $|\mathcal{F}_{valid}|$, $|\mathcal{F}_{test}|$ represent the number of quadruples in train sets, valid sets test sets, respectively. $|\mathcal{E}|$, $|\mathcal{R}|$ denote the number of entites and relations.

Dataset	$\lvert\mathcal{F}_{back}\rvert$	$\lvert\mathcal{F}_{train}\rvert$	$\lvert\mathcal{F}_{valid}\rvert$	$\lvert\mathcal{F}_{test}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Time Snapshot	Time Granularity
ICEWS14-OOG	83,448	5,772	718	705	7128	230	365	1 day
ICEWS18-OOG	444,269	19,291	2,425	2,373	23033	256	304	1 day
ICEWS0515-OOG	448,695	10,115	1,271	1,228	10488	251	4017	1 day

Table 8: Statistics of datasets for few-shot inductive TKG reasoning. $|\mathcal{F}_{train}|$, $|\mathcal{F}_{valid}|$, $|\mathcal{F}_{test}|$ represent the number of quadruples containing unseen entities in train sets, valid sets and test sets, respectively. $|\mathcal{F}_{back}|$ denotes the number of remaining quadruples without unseen entities.

2022), KGT5 (Saxena et al., 2022) and KGT5-context (Kochsiek et al., 2023). For few-shot inductive TKG reasoning, we include 1) traditional KGR methods BiQUE (Guo and Kok, 2021); 2) traditional TKGR methods, i.e. TELM (Xu et al., 2021) and TeRo (Xu et al., 2020); 3) inductive KGR methods, i.e. MEAN (Hamaguchi et al., 2017) and LAN (Wang et al., 2019); 4) mrta-learning-based method GEN (Baek et al., 2020).

A.3 Evaluation Metrics

In the experiments, we report the widely used metrics MRR (Mean Reciprocal Rank) and Hits@N to evaluate the performance of ChapTER under both two settings. MRR measures the average reciprocal ranks of all test triples. Hits@N ($N \in \{1,3,10\}$) calculates the proportion of correct entities ranked among the top-N. For fair comparison, we calculate the model results under the time-aware filtered setting (Li et al., 2022b), which only filters out the quadruples that occur at the query time. All metrics are computed by averaging over head and tail entity prediction, and model is selected by MRR value on the validation set.

A.4 Hyperparameters

We perform grid-search on hyperparameters including learning rate, prompt length, in-batch negatives, queued negative batches, train epoch and max token number. The optimal hyperparameters are summarized in Table 9.

Hyperparameters	Values
Learning rate	{1e-5, 3e-5, 5e-4, 5e-3}
Prompt Length	{2, 4, 6, 10, 15, 20, 50}
In-batch negatives	{32, 64, 128, 256, 512, 764, 1024}
Queued negative batches	{1, 2, 4}
Train epoch	{10, 15, 20}
Max token number	{50, 60, 70}

Table 9: Details of hyperparameters.

A.5 Implementation of PLM-based Baselines

We reimplement SimKGC[3], KGT5[4] and KGT5-context[5] based on their official codes. To adapt them to TKG datasets, we modify their input format from triplet to quadruplet by concatenating timestamps with their corresponding input texts. For example, a tail-prediction query input text in KGT5 can be formulated as "predict tail: 2014-01-01 | Benjamin Netanyahu | Sign formal agreement". Following their default hyperparameter setting, the hyperparameters are slightly different on different TKG datasets. For evaluation, we changed their filter setting into time-aware filter setting to align with other TKGR models.

B More Comparative Study Results

We report more transductive TKGR results on WIKI (Leblay and Chekol, 2018b) and

[3] https://github.com/intfloat/simkgc
[4] https://github.com/apoorvumang/kgt5
[5] https://github.com/uma-pi1/kgt5-context

Model	WIKI				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Graph-Based Methods								
DistMult	.109	.089	.110	.168	.120	.102	.123	.149
ComplEx	.245	.197	.273	.348	.121	.104	.124	.148
TTransE	.293	.217	.344	.424	.057	.014	.090	.112
TA-DistMult	.445	.399	.487	.517	.115	.102	.119	.139
DE-SimplE	.454	.426	.477	.496	.117	.107	.121	.135
TNTComplEx	.450	.400	.493	.520	.120	<u>.111</u>	.121	.136
CyGNet	.339	.291	.361	.419	.125	.110	.127	.148
PLM-Based Methods								
SimKGC	<u>.505</u>	<u>.439</u>	<u>.540</u>	<u>.618</u>	<u>.140</u>	.096	<u>.142</u>	<u>.200</u>
ChapTER	**.534**	**.463**	**.570**	**.653**	**.148**	**.115**	**.144**	**.202**

Table 10: Transductive TKG reasoning performance (with time-aware metrics) on WIKI and YAGO. The best results are in **bold** and the second best results are <u>underlined</u>. For fair comparison, we add corresponding timestamps of quadruples into the input text for PLM-based baselines, to equip them with the capacities of modeling time information.

YAGO (Mahdisoltani et al., 2015) in Table 10, since these datasets hold different distributions from ICEWS datasets.

C More Ablation Results

C.1 Ablation on Training Strategy

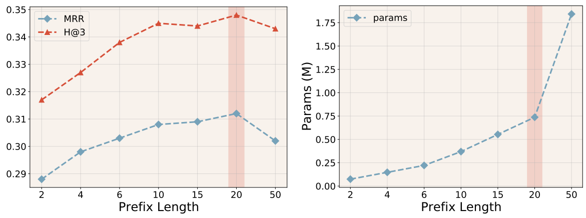
We empirically evaluate the impact of different tuning layers on ChapTER. Table 11 Lines 2-5 summarize the performance of ChapTER with different layers tuned. It can observed that ChapTER performs better than the fully tuned model, we argue that 1) ChapTER with time prefixes are effective to capture temporal information; 2) more tuning parameters may lead to an over-fitting on textual information and overlook the temporal correlations; 3) ChapTER on ICEWS14-OOG dataset is less sensitive to the changes of tuning parameters, because it is more crucial to model textual information for unseen entities in inductive TKGR. Besides, comparing line 8 and line 10, we can find that tuning bottom layers tends to obtain a better performance than tuning top layers. This could be because lower layer in Bert can capture low-level semantic features, which is more important for TKGR tasks.

C.2 Ablation on Prefix Length

We conduct extensive experiments on the impact of prefix length for ChapTER and results are shown in Figure 4. As evidenced, model performance exhibits a slight positive correlation to prefix length as increasing from 2 to 20 (MRR from .288 to .312), while number of trainable parameters is also expanding (from 0.074M to 0.737M). We can also observe that a further increase of prefix length leads to an inferior performance (from .312 to .302), as additional complexity imposes considerable challenges.

No.	Model	ICEWS14		ICEWS14-OOG	
		MRR	H@10	MRR	H@10
1	ChapTER	**.332**	**.515**	**.361**	.752
2	*w/* last layer tuned	.323	.504	.347	.748
3	*w/* last 6 layers tuned	.316	.499	.359	.748
4	*w/* first layer tuned	.311	.494	.333	.721
5	*w/* fully tuned	.316	.493	.357	.748

Table 11: Performance of ChapTER with different training strategies. Lines 2-5 report variants on training strategy.



(a) MRR & H@3 Score (b) Trainable Parameters

Figure 4: Impact (performance and parameter size) of prefix length on ICEWS14 dataset.