Bring Invariant To Variant: A Contrastive Prompt-based Framework for Temporal Knowledge Graph Forecasting

Ying Zhang[1,*], **Xinying Qian**[1], **Yu Zhao**[1], **Baohang Zhou**[1], **Kehui Song**[1], **Xiaojie Yuan**[1]
[1]College of Computer Science, VCIP, TMCC, TBI Center, Nankai University, China
{yingzhang, yuanxj}@nankai.edu.cn
{qianxinying, zhaoyu, zhoubaohang, songkehui}@dbis.nankai.edu.cn

Abstract

Temporal knowledge graph forecasting aims to reason over known facts to complete the missing links in the future. Existing methods are highly dependent on the structures of temporal knowledge graphs and commonly utilize recurrent or graph neural networks for forecasting. However, entities that are infrequently observed or have not been seen recently face challenges in learning effective knowledge representations due to insufficient structural contexts. To address the above disadvantages, in this paper, we propose a **Co**ntrastive **P**rompt-based framework with **E**ntity background information for **T**KG forecasting, which we named **CoPET**. Specifically, to bring the time-invariant entity background information to time-variant structural information, we employ a dual encoder architecture consisting of a candidate encoder and a query encoder. A contrastive learning framework is used to encourage the query representation to be closer to the candidate representation. We further propose three kinds of trainable time-variant prompts aimed at capturing temporal structural information. Experiments on two datasets demonstrate that our method is effective and stays competitive in inference with limited structural information. Our code is available at https://github.com/qianxinying/CoPET.

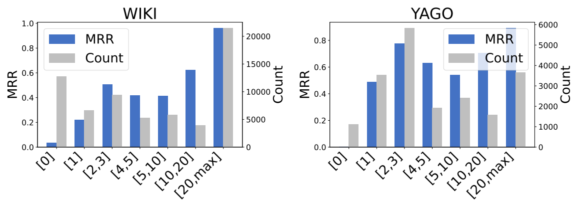
Keywords: temporal knowledge graph forecasting, pre-trained language models, contrastive learning

1. Introduction

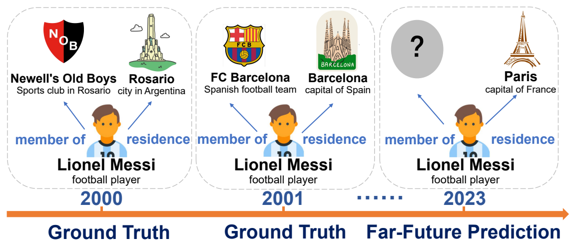
Knowledge Graphs(KGs) are structured representations of knowledge in the form of a triplet *(subject, predicate, object)*. Various knowledge graphs have been constructed and widely applied in downstream applications, such as recommendation systems (Guo et al., 2020) and question answering (Zhang et al., 2018). However, in the real world, some facts change over time. For example, the fact *(Lionel Messi, member of, FC Barcelona)* is invalid after Messi announced his departure from FC Barcelona in 2021. Therefore, some KGs store time-aware facts or events as quadruple *(subject, predicate, object, timestamp)*, and such KGs are called temporal knowledge graphs(TKGs).

Since existing TKGs remain incomplete, given a TKG with timestamps ranging from t_0 to t_n, TKG forecasting task(extrapolation) (Jin et al., 2019) aims at predicting missing entities in future timestamps $t > t_n$, which is more challenging and less-explored than completing the facts in observed timestamps(interpolation) (García-Durán et al., 2018) between t_0 and t_n.

To mine the changes in TKGs over time, some methods (Jin et al., 2019) leverage recurrent neural networks (RNNs) (Zaremba et al., 2014) or graph neural networks (GNNs) (Zhou et al., 2020) to explore the topological relations among entities in TKGs and the temporal dependencies among facts. Meanwhile, some methods (Xu et al., 2023) incorporate pre-trained language models(PLMs), but they still rely on the temporally adjacent facts to model the representations. Thus, they face challenges with limited structural information:



(a) The count of entities neighbors and MRR results on WIKI and YAGO datasets of RE-NET.



(b) An example of far-future prediction with entity background information.

Figure 1: Two challenges in TKG forecasting task with limited structural information.

(1) Long-tailed entities prediction is to predict entities with few neighbors. There are many long-tailed entities in KGs (Tan et al., 2023) that have insufficient neighbors. As illustrated in Figure 1a, on the WIKI dataset, approximately 20.12%

[*] Corresponding author.

LREC-COLING 2024, pages 2526–2536
20-25 May, 2024.

Method	Multi-step Inference	Future Prediction	Inductive Prediction	Reasoning Information			
				time	structure	text	description
BoxTE (Interpolated)	✓	×	×	✓	✓	×	×
Hismatch (GNN-based)	×	✓	×	✓	✓	×	×
PPT (PLM-based)	✓	✓	✓	✓	✓	✓	×
SimKGC (static PLM-based)	✓	×	✓	×	×	✓	✓
CoPET	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of our model CoPET with other models in terms of their capabilities in multi-step inference, future prediction, inductive prediction, and information utilized during reasoning.

of entities have no neighbors, and 53.93% have fewer than 5 neighbors. However, existing models primarily learn knowledge representations from historical temporal structures. Consequently, their performance tends to deteriorate as the number of neighboring entities decreases. For example, the traditional structured-based method RE-NET (Jin et al., 2019) struggles to infer answers when entities have no neighbors, as shown in Figure 1a.

(2) Far-future prediction is a significant task that has not received much attention so far. As shown in Figure 1b, far-future prediction refers to forecasting events over extended temporal horizons, posing particular challenges due to the limited structural information derived from recent subgraphs. Moreover, traditional methods often focus on modeling the subgraph sequence relevant to the query, which could result in a gradual decline in reasoning abilities over time (Han et al., 2021).

Hence, it is insufficient for existing methods to solely incorporate structural information while disregarding the entity background information, particularly when confronted with the challenges mentioned above. For instance, in Figure 1b, predicting which team *Messi* would play for in 2023 can be challenging if we solely rely on historical information. However, by considering the neighbor information (*Messi* resides in *Paris* in 2023) and leveraging the background information of the entity (*Paris* Saint-Germain Club: a football club in *Paris*), we can easily infer the answer. Therefore, we propose to bring time-invariant entity background information to time-variant structural information.

In this paper, we introduce CoPET, a **Co**ntrastive **P**rompt-based framework with **E**ntity background information for **T**KG forecasting. Specifically, we treat the TKG forecasting task as a matching process between time-variant queries and time-invariant candidates. A dual-encoder architecture is employed to model the two types of embeddings separately. First, we align the entities with their corresponding Wikidata pages to obtain entity descriptions and names. Whenever an entity appears, we concatenate its name with its description to form the time-invariant entity background information. We further introduce three trainable prompts, namely predicate prompts, time prompts, and neighbor prompts, to learn temporal structural information in queries. Considering the repeatability of historical events, we propose a history-based re-ranking strategy to improve the accuracy. Experiments on two datasets demonstrate the effectiveness of CoPET. More experiments prove that CoPET stays competitive in inference with limited structural information. Overall, our work makes the following contributions:

- To the best of our knowledge, we are the first to incorporate time-invariant entity background information for the TKG forecasting task and address two challenges with limited structural information in existing methods.
- We propose a contrastive prompt-based framework for the TKG forecasting task, CoPET. We design templates with time-invariant entity background information and three kinds of time-variant prompts for TKGs.
- Experiments on two datasets demonstrate that CoPET is effective and stays competitive with limited structural information.

2. Related work

2.1. Temporal Knowledge Graph Reasoning

The interpolation setting of TKG reasoning aims to complete missing events at past timestamps. TTransE (Jiang et al., 2016) extends the idea of TransE (Bordes et al., 2013) by incorporating temporal order information among facts. TA-TransE and TA-DistMult (García-Durán et al., 2018) are also extended versions of TransE and DistMult (Yang et al., 2014) that incorporate the temporal embeddings. DE-SimplE (Goel et al., 2020) applies the general diachronic embedding function to obtain the entity representation. BoxTE (Messner et al., 2022) extends from the KGC model BoxE (Abboud et al., 2020) via the relation-specific transfer matrix. However, these approaches cannot obtain the representations of the unseen timestamps and are unsuitable for the TKG forecasting task.

For the TKG forecasting task, RE-NET (Jin et al., 2019) applies GRU to transmit information of past event sequences sequentially and uses GCN as the aggregator. CyGNet (Zhu et al., 2021) adopts a copy-generation mechanism to identify repetitive events. TANGO (Han et al., 2021) employs neural ordinary differential equations to model the TKGs. These methods rely on historical structural information for reasoning, leading to poor performance on long-tailed entities. Some recent studies focus on the single-step inference that utilizes all the ground truth quadruples in train/valid/test set before the current timestamp, TITer (Sun et al., 2021) adopts reinforcement learning and designs a novel time-shaped reward based on Dirichlet distribution. Hismatch (Li et al., 2022) applies two structure encoders to capture the information contained in the historical structures of the query and candidate entities. However, they could not address the far-future prediction challenge since they rely on closer ground truth. In contrast, we focus on multi-step inference, utilizing only the quadruples before the timestamp of the train set.

2.2. Pre-trained Language Models for Knowledge Graph Reasoning

Pre-trained Language Models(PLMs), including BERT (Devlin et al., 2018), GPT (Liu et al., 2021), are first pre-trained on massive amounts of unlabeled text corpora and then fine-tuned. They have been increasingly prevalent in NLP, but lack domain-specific knowledge (Yao et al., 2019).

Due to their ability to capture context information, PLM-based models have been considered for knowledge graph reasoning in recent years. KG-BERT (Yao et al., 2019) directly concatenates subject, predicate, and object from triples as input to PLMs, MTL-KGC (Choi et al., 2021) introduces multi-task learning, and StAR (Wang et al., 2021) integrates structured knowledge into text encoders. These methods are still inferior to structure-based methods. To improve the performance of the PLM-based method, SimKGC (Wang et al., 2022a) and LMKE (Wang et al., 2022b) adopt contrastive learning, PKGC (Lv et al., 2022) converts each triple into natural prompt sentences with templates. CS-PromKG (Chen et al., 2023) employs conditional soft prompts, which are generated by the embeddings of entities and relations. However, these methods are only applicable to static settings or interpolation tasks, and thus cannot be directly employed in TKG forecasting tasks. SToKE (Gao et al., 2023) transforms structural and temporal contexts into a structured event sequence. But it only focuses on interpolated settings. ECOLA (Han et al., 2022) is the first to enhance temporal knowledge embedding with temporally textual information. Nevertheless, acquiring descriptions for each quadruple over time is difficult. PPT (Xu et al., 2023) utilizes the PLMs masked token prediction task to solve the TKG forecasting task. However, it relies on temporally adjacent facts information to predict future facts without considering entity background information of entities. CoPET, on the other hand, combines time-invariant entity background information and time-variant structural information, effectively tackling both challenges under the condition of limited structural information. Table 1 gives a comparison of the most relevant related works.

3. Method

3.1. Preliminaries

Temporal knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{P}, \mathcal{T}, \mathcal{Q}\}$ is a directed graph where vertices are a set of entities $\mathcal{E}$, and the edges are a set of predicates $\in \mathcal{P}$ with timestamps $\mathcal{T}$. The quadruple set $\mathcal{Q} = \{(s, p, o, t)\} \subseteq \mathcal{E} \times \mathcal{P} \times \mathcal{E} \times \mathcal{T}$ represents the time-variant facts, where s and o are subject entity and object entity, p is predicate between s and o at timestamp t.

The TKG forecasting task aims at completing the missing facts of future times. Given a particular query $(s_q, p_q, ?, t_q)$, TKG forecasting model predicts the object entity with a series of facts known before t_q: $\{(s, p, o, t)|(s, p, o, t) \in \mathcal{Q}, t < t_q\}$. For each quadruple (s, p, o, t), we add its inverse quadruple (o, p^{-1}, s, t) for subject prediction, where p^{-1} represents the inverse predicate of p.

3.2. Overview

Figure 2 shows our proposed framework CoPET, a Contrastive Prompt-based framework for TKG forecasting. We adopt the bi-encoder architecture that encodes time-variant queries $(s, p, ?, t)$ and candidate entities $e \in \mathcal{E}$ with Query Encoder and Candidate Encoder, respectively. The two encoders are initialized with the same PLMs and do not share parameters. The score function $\phi(s, p, e, t)$ calculates the similarity between query embedding and candidate embedding as the plausibility of the quadruple.

In CoPET, we aim to bring time-invariant entity background information to time-variant knowledge representations to solve the challenges with limited structural information. Thus, we first design the templates for queries and candidates to obtain the time-invariant entity background information. Moreover, we transform the temporal structures of TKGs into trainable time-variant prompts and add the prompts into query templates. During the training phase, CoPET utilizes contrastive loss to efficiently distinguish the knowledge representations in different timestamps. In the inference phase, CoPET first calculates the matching score between

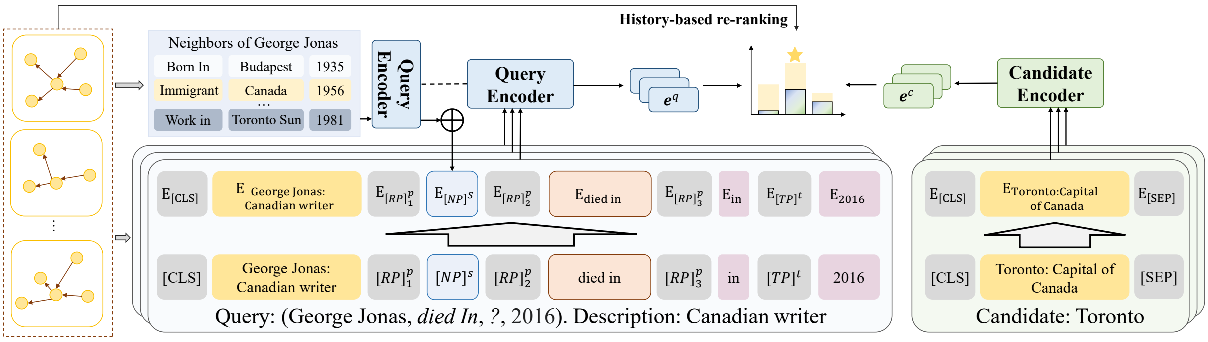


Figure 2: The overall architecture of CoPET. Query Encoder takes the query along with time-invariant entity background information and time-variant prompts as input, and Candidate Encoder takes the time-invariant information of candidate entities as input. Cosine similarity is used to calculate the score between the outputs. Based on the history of TKGs, we utilize a re-ranking strategy.

query embedding and candidate embedding. Then, a history-based re-ranking strategy is adopted to capture history repeatability.

3.3. Query Encoder

The Query Encoder aims at encoding both the temporal structure information of the query $(s, p, ?, t)$ and the time-invariant context of subject entity s. To this end, we propose to transform both information into natural language sequences and design templates to combine them in the input of PLMs.

We first convert the query into natural language sequences as $T_q(s, p, t)$ ="`[CLS][subject: description][predicate][timestamp]`", where the entity descriptions are concatenated with the subject name. Take Figure 2 as an example, the template of the query *(George Jonas, died In, ?, 2016)* would be "`[CLS]` *George Jonas: Canadian writer* `[SEP]` *died in* `[SEP]` *in 2016*.".

We obtain the query embeddings from the projected hidden state of `[CLS]` token of query sequences as $\mathbf{e}_q = QueryEncoder(T_q(s, p, t)) = \mathbf{W}_q \mathbf{h}_q^{\texttt{[CLS]}}$, where $\mathbf{W}_q \in \mathbb{R}^{emb \times d}$ projects the hidden size of PLMs d to embedding size emb.

Moreover, motivated by the progress of prompt learning (Lester et al., 2021), we further transform the temporal structural information in TKGs into three types of trainable soft prompts and insert the prompts in the template. The token embeddings of predicate prompts and time prompts are randomly initialized, whereas the neighbor prompts are initialized using the mean pooling of sampled relational neighbor embeddings. We discuss the effect of templates with varying orders in Section 4.4.4.

Predicate prompts. For each relational predicate $p \in \mathcal{P}$, we add three soft prompts into vocabulary as special tokens $\{\texttt{[RP]}_i^p\}_{i=1}^3$. Then we replace the `[SEP]` in the template with the three special tokens following PKGC (Lv et al., 2022).

Time prompts. Similar to predicate prompts, for each timestamp $t \in \mathcal{T}$, we add a special token $\texttt{[TP]}^t$ and insert it between *in* and `[timestamp]`.

Neighbor prompts. The association between the subject entity and its relational neighbors usually implies a possible object entity and better represents the query. For capturing the neighboring context in TKGs, we propose neighbor prompt $\texttt{[NP]}^s$ for the subject entity, whose token embedding is replaced by the mean pooling of sampled neighbor query representations of s as shown in Equation (1). The neighbor set of s at time t is $\mathcal{R}^s = \{(p', o', t') | (s, p', o', t') \in \mathcal{Q}, t' < t\}$. It consists of both in and out neighbors of s since we add the inverse quadruple in TKGs.

$$\mathbf{e}_{\texttt{[NP]}^s} = \frac{1}{|\mathcal{R}^s|} \sum_{(p',o',t') \in \mathcal{R}^s} QueryEncoder(T_q(p', o', t')) \tag{1}$$

3.4. Candidate Encoder

Candidate Encoder aims at encoding the candidate entities $e \in \mathcal{E}$ with the time-invariant context. So even with less structural information, i.e. long-tailed entity or far-future prediction, CoPET could still learn entity representations from the entity background information of the entity.

The template for the candidate entity is $T_c(e)$ = "`[CLS] [entity: description]. [SEP]`". We also obtain the candidate embeddings from the projected hidden states of `[CLS]` token of candidate sequence as $\mathbf{e}_c = CandidateEncoder(T_c(c)) = \mathbf{W}_c \mathbf{h}_c^{\texttt{[CLS]}}$, where $\mathbf{W}_c \in \mathbb{R}^{emb \times d}$ projects the hidden size of PLMs d to embedding size emb.

3.5. Training

We utilize InfoNCE loss with additive margin (Chen et al., 2020; Yang et al., 2019) to train the model as shown in Equation (2).

$$\mathcal{L} = -\log \frac{e^{(\phi(s,p,o,t)-\gamma)/\tau}}{e^{(\phi(s,p,o,t)-\gamma)/\tau} + \sum_{o' \in \mathcal{N}^o} e^{\phi(s,p,o',t)/\tau}} \quad (2)$$

$$\phi(s,p,o,t) = \cos(\mathbf{e}_q, \mathbf{e}_c) = \frac{\mathbf{e}_q \cdot \mathbf{e}_c}{||\mathbf{e}_q|| \times ||\mathbf{e}_c||} = \mathbf{e}_q \cdot \mathbf{e}_c \quad (3)$$

We adopt cosine similarity similarity as score function $\phi(s,p,o,t) = \cos(\mathbf{e}_q, \mathbf{e}_c)$. Due to the embeddings $\mathbf{e}_q$ and $\mathbf{e}_c$ are L2 normalized, the cosine similarity is simplified as the dot product between them as shown in Equation (3). For efficient training, we simply apply *in-batch negative sampling* strategy and the other candidate entities o' in batch besides the ground truth entity o is seen as the negative set $\mathcal{N}^o$ of the quadruple. The contrastive loss with additive margin $\gamma > 0$ encourages the model to maximize the similarity between the query and the ground truth entity $\phi(s,p,o,t)$. The temperature parameter τ adjusts the smoothness of the score distribution. γ and τ are hyper-parameters.

3.6. Inference

With the query embedding $\mathbf{e}_q$ and the candidate embedding $\mathbf{e}_c$, the score function $\phi(s,p,o,t)$ calculates the similarity between the two embeddings as the plausibility of the quadruple. We adopt the cosine similarity as shown in Equation (3). For a particular query $(s,p,?,t)$, we calculate the similarity distribution $F(s,p,t) \in \mathbb{R}^{|\mathcal{E}|}$ between the query and all the entities $c \in \mathcal{E}$, and the candidate with the highest score is regarded as the predicted entity as shown in Equation (4).

$$\operatorname*{argmax}_c F(s,p,t) = \operatorname*{argmax}_c [\phi(s,p,c,t)], c \in \mathcal{E} \quad (4)$$

In terms of inference time complexity, considering BERT as an example, each query $(s,p,?,t)$ in the test set $\mathcal{Q}_{test}$ requires $O(2 \times 2 \times |\mathcal{Q}_{test}|)$ BERT forward passes computations: one for computing the embeddings of its neighbors, and another for computing the embeddings of the query itself. In total, CoPET requires $O(4 \times |\mathcal{Q}_{test}| + O(\mathcal{E}))$ BERT forward passes computations.

3.7. History-based Re-ranking

Considering the repeatability of historical events (Zhu et al., 2021), we propose a re-ranking strategy based on the history of entities to increase the score of entities that have appeared before and minimize the scores of unrelated entities.

Specifically, we build a history vocabulary $H(s,p,t)$ for each query$(s,p,?,t)$, which is a $|\mathcal{E}|$-dimensional multi-hot vector. Each element of $H(s,p,t)$ is the frequency of the entity e that $(s,p,e,t') \in \mathcal{Q}, t' < t$.

For modeling history while preserving the similarity distribution from knowledge representations, instead of enlarging the probability of the *seen entity* in $H(s,p,t)$, we decrease the probability of the *unseen entity* in history that are less related to (s,p). We use $\dot{H}(s,p,t)$ to convert the unseen entity frequency in $H(s,p,t)$ into a small negative number while the seen entity into 0. By adding the similarity scores $F(s,p,t)$ and $\dot{H}(s,p,t)$ to get $F'(s,p,t) \in \mathbb{R}^{|\mathcal{E}|}$ as shown in Equation (5), we can minimize the probability of the unrelated entities based on their history.

$$F'(s,p,t) = F(s,p,t) + \dot{H}(s,p,t) \quad (5)$$

Dataset	YAGO11k	Wikidata12k
# Train	161,286	539,286
# Valid	19,523	67,538
# Test	20,026	63,110
# Entity	10,623	12,554
# Relation	10	24
Timegap	1 year	1 year

Table 2: The statistical information of datasets for TKG forecasting task.

4. Experiments

4.1. Experimental Settings

Datasets. Our model is evaluated on two public TKG datasets: Wikidata12k (WIKI) (Leblay and Chekol, 2018) and YAGO11k (YAGO) (Mahdisoltani et al., 2014). Both WIKI and YAGO are knowledge bases that store facts with corresponding timestamps. We use subsets of these datasets that have a time granularity of years. We adopt the same dataset split strategy as presented in (Jin et al., 2019) and split the dataset into train/valid/test by timestamps, resulting in train time < valid time < test time. To obtain time-invariant entity descriptions for WIKI and YAGO, we align each entity with its corresponding Wikidata page and extract the description section as the entity's description. Table 2 shows the dataset statistics.

Evaluation Metrics. We utilize standard metrics in link prediction tasks to evaluate our model: mean reciprocal rank (MRR) and Hits@k (k$\in \{1,3,10\}$). MRR is the average reciprocal rank of all test triples. Hits@k calculates the proportion of correct entities ranked among the top-k. MRR and Hits@k are reported under the time-aware filtered setting (Han et al., 2020), which ignores the quadruples occurring at the query time. All metrics are computed by averaging over two ways: subject prediction $(?,p,o,t)$ and object prediction $(s,p,?,t)$.

Hyperparameters. We initialize both encoders with bert-base-uncased (English). We choose AdamW (Kingma and Ba, 2014) as our optimizer.

The learning rate is set as 1e-5. The temperature value τ is initialized as 0.05, and the additive margin for InfoNCE loss is 0.02. We sampled the 5 neighbors closest to the query timestamps. Models are trained with batch size 256 and in 10 epochs.

Baselines. We compare our model with three types of KG reasoning models, static models, temporal models under interpolation setting and extrapolation setting, and PLM-based models. (1) Static models: DistMult (Yang et al., 2014), TuckER (Balažević et al., 2019), CompGCN (Vashishth et al., 2019). (2) Temporal interpolation models: TTransE (Jiang et al., 2016), TA-DistMult (García-Durán et al., 2018), DE-SimplE (Goel et al., 2020), TNT-ComplEx (Lacroix et al., 2020). (3) Temporal extrapolation models: RE-Net (Jin et al., 2019), CyGNet (Zhu et al., 2021), TANGO-TuckER (Han et al., 2021). (4) PLM-based models: SimKGC (Wang et al., 2022a), CS-PromKG (Chen et al., 2023), ECOLA-DyERNIE (Han et al., 2022), PPT (Xu et al., 2023). For SimKGC and CS-PromKG, as they are static models, we follow the method in Chen et al. (2023), adding timestamp into its input.

4.2. Main Result

We compare CoPET with baselines in Table 3. We reuse the baseline results in group (1)-(3) reported by Han et al. (2021).

(1) Among static methods, CoPET outperforms all static models, emphasizing the significance of incorporating temporal information in TKGs.

(2) Interpolation methods such as TTransE, TA-DistMult, DE-SimplE, and TNT-ComplEx exhibit lower performance on the two datasets because they lack the capability to proficiently capture and reason about future temporal dependencies.

(3) In comparison to extrapolation methods, CoPET surpasses RE-NET, CyGNet, and TANGO-TuckER on both datasets. This can be attributed to the limited structural information available in the WIKI and YAGO datasets. The relatively small number of neighbors for entities in these datasets hinders the effectiveness of structural methods. However, by bringing time-invariant entity background information, CoPET achieves better results.

Compared with (4) PLM-based methods, CoPET also outperforms the KGC models SimKGC, CS-PromKG, and the TKGC models ECOLA-DyERNIE and PPT on both datasets. For the KGC models SimKGC and CS-PromKG, just adding timestamps into their input has already achieved good results, which shows that PLMs can bring improvement in TKGC. The superiority of CoPET can be attributed to its effective utilization of time information. For the TKGC models ECOLA-DyERNIE and PPT, we achieve better results than them. These methods do not consider time-invariant information during modeling, causing their representations to change over time. CoPET, on the other hand, combines time-invariant entity background information and time-variant structural information and therefore achieves better results.

4.3. Ablation Study

To further analyze how each component of CoPET contributes to the final results, we conduct ablation studies on all datasets under the same settings. The results are shown in Table 4.

Effect of entities description. We first remove the description for each entity to investigate the impact, the result shows a substantial decrease in performance on both datasets. This suggests that time-invariant entity description provides crucial background knowledge that significantly enhances the reasoning ability of CoPET.

Effect of soft prompts. We replace all the predicate prompts and time prompts with `[SEP]` token. After removing prompts, the results significantly decreased, indicating that soft prompts can enhance the learning of knowledge representations.

Effect of neighbor prompts. After removing the neighbor prompts, the results have also significantly declined. This demonstrates the importance of considering temporal structure information in TKGs.

Effect of re-ranking strategy. We utilize a re-ranking strategy that decreases the probability of unrelated entities based on the history of their occurrences in previous timestamps. The results indicate that the history-based re-ranking strategy enhances the performance of CoPET. This improvement could be attributed to the historical repeatability across TKGs, which assists in filtering out unrelated candidates in future timestamps.

4.4. Further Discussion

To further analyze the superior performance of CoPET compared to other models, we conduct a series of experiments to gain further insights.

4.4.1. Representations Visualization

To investigate the ability of CoPET to capture and distinguish entity information of distinct temporal facts, we visualize the candidate representations and query representations as shown in Figure 3.

For candidate representations, we visualize the representations from 5 categories, with 50 entities randomly selected from each category. The Figure 3a reveals that entities belonging to the same category are tightly clustered together, indicating that the candidate encoder effectively learns high-quality time-invariant representations.

For query representations, we visualize the representations of the subject entity and predicate pairs

Model	WIKI				YAGO			
	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@10 ↑	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@10 ↑
DistMult (2015)	49.66	46.17	52.81	54.13	54.84	47.39	59.81	68.52
TuckER (2019)	50.01	46.12	53.60	54.86	54.86	47.32	59.63	68.96
CompGCN (2019)	49.88	45.78	52.91	55.58	54.35	46.72	59.26	68.29
TTransE (2016)	29.27	21.67	34.43	42.39	31.19	18.12	40.91	51.21
TA-DistMult (2018)	44.53	39.92	48.73	51.71	54.92	48.15	59.61	66.67
DE-SimplE (2020)	45.43	42.6	47.71	49.55	54.91	51.64	57.30	60.17
TNTComplEx (2020)	45.03	40.04	49.31	52.03	57.98	52.92	61.33	60.69
RE-Net (2021)	49.66	46.88	51.19	53.48	58.02	53.06	61.08	66.29
CyGNet (2021)	33.89	29.06	36.10	41.86	52.07	45.36	56.12	63.77
TANGO-TuckER (2021)	51.60	49.61	52.45	54.87	<u>62.50</u>	<u>58.77</u>	<u>64.73</u>	<u>68.63</u>
SimKGC (2022)	47.28	40.22	51.93	56.70	51.51	43.74	54.45	65.22
CS-PromKG (2023)	34.00	26.43	37.40	46.49	46.49	38.10	52.19	60.88
ECOLA-DyERNIE (2022)	41.22	33.02	45.00	-	-	-	-	-
PPT (2023)	<u>53.95</u>	<u>50.05</u>	<u>57.28</u>	<u>60.56</u>	60.42	55.11	64.33	68.70
CoPET	**56.32**	**51.21**	**58.31**	**67.70**	**64.01**	**59.06**	**67.65**	**73.21**

Table 3: Performance (in percentage) on WIKI and YAGO. The **best** results and the <u>second-best</u> ones are highlighted. We re-implement SimKGC (Wang et al., 2022a) and PPT (Xu et al., 2023) and reuse results of other baselines in Han et al. (2021).

Model	WIKI				YAGO			
	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@10 ↑	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@10 ↑
CoPET	**56.32**	**51.21**	**58.31**	**67.70**	**64.01**	**59.06**	**67.65**	**73.21**
w/o description	49.76	43.29	52.25	57.29	51.38	45.15	53.17	61.09
w/o soft prompts	52.74	45.76	56.14	65.04	52.11	45.79	54.31	64.42
w/o neighbor prompts	51.93	44.03	55.24	65.68	52.50	45.72	55.26	65.65
w/o re-rank	<u>54.75</u>	<u>47.65</u>	<u>58.86</u>	<u>67.28</u>	<u>59.47</u>	<u>53.93</u>	<u>61.71</u>	<u>69.84</u>

Table 4: Ablation experiments results of CoPET. "descriptions" denotes the entity descriptions. "soft prompts" denotes predicate prompts and time prompts. "re-rank" denotes the re-ranking in Section 3.7

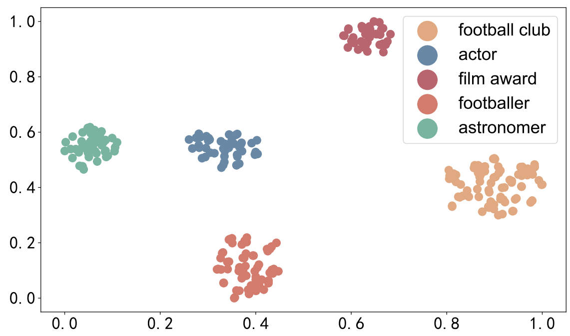
(s, p) at different times. Similarly, we randomly select 5 pairs at 50 timestamps. The Figure 3b shows that subject entities and predicate pairs from the query encoder possess distinct representations at various timestamps. However, overall, the representations are clustered together based on their semantic information, indicating that CoPET has the ability to capture both time-invariant and time-variant information.

4.4.2. Long-tailed Entities Prediction

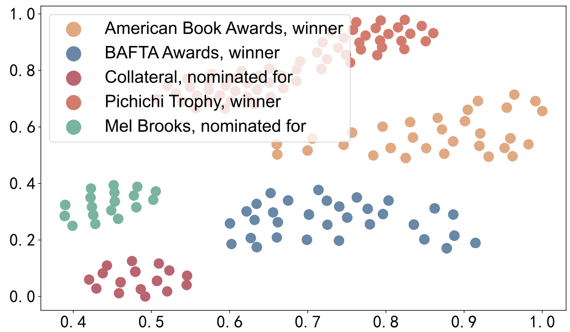
To evaluate the efficacy of our methods in handling long-tailed entities, we divide entities into five categories based on the number of their neighbors. Figure 4 shows the MRR results of CoPET on each subset compared to PLM-based methods(PPT, SimKGC) and structure-based methods(CyGNet and RE-NET). We also illustrate the count of entities in each subset in the line chart. In terms of long-tailed entities (categories with fewer than 5 neighbors), CoPET outperforms all other models. Particularly when entities have 0 neighbors, structure-based models face challenges in reasoning, while CoPET maintains its performance. However, for popular entities, structure-based methods generally exhibit superior performance. The success of CoPET can be attributed to its utilization of both time-invariant and time-variant information. When confronted with limited structural information, CoPET leverages time-invariant entity background information to facilitate reasoning, resulting in improved performance on long-tailed entities. On the other hand, when structural information is available, CoPET incorporates temporal structural information, enabling it to outperform PLM-based methods.

4.4.3. Far-Future Prediction

The reasoning ability of TKG forecasting models tends to decline over time, while some models (Li et al., 2022; Sun et al., 2021) can only perform single-step reasoning, requiring the feeding of ground truth data during testing. However, in some scenarios, it is necessary to predict far-future facts. Therefore, we regard far-future prediction as a crucial ability and evaluate the models under this setting. To analyze the model's performance under far-future prediction, we cluster the Hits@10 results based on different timestamps. Δt denotes the time interval between the current time and the last time in the training set.



(a) Candidate Encoder representations visualization. We visualize the embeddings from 5 categories, with 50 entities randomly selected from each category.



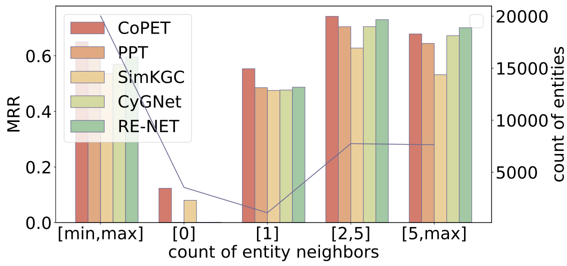
(b) Query Encoder representations visualization. We visualize the embedding of 5 subject entities and predicate pairs (s, p) at 50 time stamps.

Figure 3: Embedding visualizations from Candidate Encoders and Query Encoder on WIKI dataset with t-SNE (Van der Maaten and Hinton, 2008).

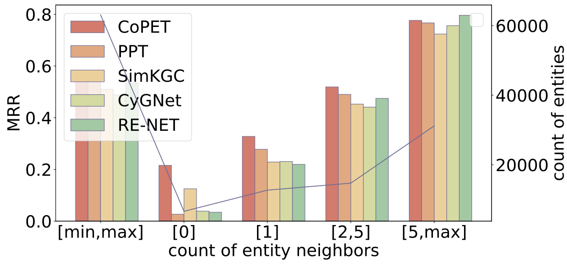
Figure 5 indicates a downward trend in Hits@10 for all models over time, which is consistent with our expectations. Although CoPET initially underperformed compared to CyGNet in both datasets, its performance eventually surpassed it, and the declining rate of CoPET is much lower compared to other models. The excellent performance of CoPET could be attributed to the incorporation of time-invariant entity background information in addition to time-variant structural information. As a result, when predicting far-future facts without the support of ground truth data of recent times, CoPET outperforms all the other methods.

4.4.4. Results of different templates

In this section, we explore the effects of templates with different soft prompt orders, particularly focusing on the positioning of time information and neighbor prompts. Table 5 shows the MRR results of different templates. Notably, positioning neighbor prompts between the subject entity and predicate, and placing time information at the end, yields the best results, which is the template used in CoPET.

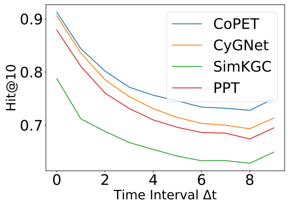


(a) Results on YAGO.

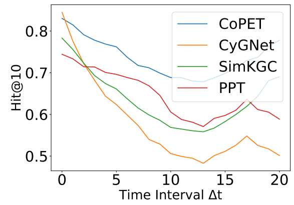


(b) Results on WIKI.

Figure 4: The MRR results of CoPET, PPT, SimKGC, and CyGNet in different counts of entity neighbors. The line graph displays the number of quadruples in each category.



(a) Results on YAGO.



(b) Results on WIKI.

Figure 5: The Hits@10 results of CoPET, PPT, SimKGC, and CyGNet in different time intervals.

5. Conclusion

In this work, we propose to bring the time-invariant entity background information to time-variant structural information and introduce a prompt-based contrastive temporal knowledge representation learning framework with entity background information named CoPET. To encourage the query representation to be closer to the candidate representation, we employ a dual encoder architecture with a contrastive learning framework. For time-invariant information, we obtain entity descriptions by aligning each entity with Wikidata and concatenating its description without considering the time. For time-variant information, we introduce three trainable time-variant prompts, namely predicate prompts, time prompts, and neighbor prompts, to learn temporal structural information. Extensive experimental results reveal that our model outperforms state-of-the-art baselines and gains better performances with limited structural information.

Template	MRR
subject `[Neighbor]` predicate `[Time]`	**59.47**
`[Time]` subject predicate `[Neighbor]`	<u>58.90</u>
subject predicate `[Time]` `[Neighbor]`	58.61
subject predicate `[Neighbor]` `[Time]`	58.16
`[Time]` subject `[Neighbor]` predicate	57.99

Table 5: Different MRR results of different templates on YAGO datasets.

6. Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 62272250), and the Natural Science Foundation of Tianjin, China (No. 22JCJQJC00150).

7. Bibliographical References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. 2023. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Bonggeun Choi, Daesik Jang, and Youngjoong Ko. 2021. Mem-kgc: Masked entity model for knowledge graph completion with pre-trained language model. *IEEE Access*, 9:132025–132032.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yifu Gao, Yongquan He, Zhigang Kan, Yi Han, Linbo Qiao, and Dongsheng Li. 2023. Learning joint structural and temporal contextualized knowledge embeddings for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 417–430.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.

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

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. xerte: Explainable reasoning on temporal knowledge graphs for forecasting future links. *arXiv preprint arXiv:2012.15537*.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364.

Zhen Han, Ruotong Liao, Jindong Gu, Yao Zhang, Zifeng Ding, Yujia Gu, Heinz Köppl, Hinrich Schütze, and Volker Tresp. 2022. Ecola: Enhanced temporal knowledge embeddings with contextualized language representations. *arXiv preprint arXiv:2203.09590*.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. 2016. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724.

Woojeong Jin, He Jiang, Meng Qu, Tong Chen, Changlin Zhang, Pedro Szekely, and Xiang Ren. 2019. Recurrent event network: Global structure inference over temporal knowledge graph.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.

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*, pages 3045–3059.

Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022. Hismatch: Historical structure matching based temporal knowledge graph reasoning. *arXiv preprint arXiv:2210.09708*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. Association for Computational Linguistics.

Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7779–7787.

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

Zhaoxuan Tan, Zilong Chen, Shangbin Feng, Qingyue Zhang, Qinghua Zheng, Jundong Li, and Minnan Luo. 2023. Kracl: contrastive learning with graph context modeling for sparse knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 2548–2559.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022a. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167*.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022b. Language models as knowledge embeddings. *arXiv preprint arXiv:2206.12617*.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023. Pre-trained language model with prompts for temporal knowledge graph completion. *arXiv preprint arXiv:2305.07912*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5370–5378. International Joint Conferences on Artificial Intelligence Organization.

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

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

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.

8. Language Resource References

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018*, pages 1771–1776.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference.