Learning to Sample and Aggregate: Few-shot Reasoning over Temporal Knowledge Graphs

Ruijie Wang, Zheng Li, Dachun Sun, Shengzhong Liu, Jinning Li, Bing Yin, Tarek Abdelzaher

1University of Illinois at Urbana Champaign, IL, USA
2Amazon.com Inc, CA, USA
{ruijiw2, dsun18, sl29, jinning4, zaher}@illinois.edu
{amzzhe, alexbyin}@amazon.com

Abstract

In this paper, we investigate a realistic but underexplored problem, called few-shot temporal knowledge graph reasoning, that aims to predict future facts for newly emerging entities based on extremely limited observations in evolving graphs. It offers practical value in applications that need to derive instant new knowledge about new entities in temporal knowledge graphs (TKGs) with minimal supervision. The challenges mainly come from the few-shot and time shift properties of new entities. First, the limited observations associated with them are insufficient for training a model from scratch. Second, the potentially dynamic distributions from the initially observable facts to the future facts ask for explicitly modeling the evolving characteristics of new entities. We correspondingly propose a novel Meta Temporal Knowledge Graph Reasoning (MetaTKGR) framework. Unlike prior work that relies on rigid neighborhood aggregation schemes to enhance low-data entity representation, MetaTKGR dynamically adjusts the strategies of sampling and aggregating neighbors from recent facts for new entities, through temporally supervised signals on future facts as instant feedback. Besides, such a meta temporal reasoning procedure goes beyond existing meta-learning paradigms on static knowledge graphs that fail to handle temporal adaptation with large entity variance. We further provide a theoretical analysis and propose a temporal adaptation regularizer to stabilize the meta temporal reasoning over time. Empirically, extensive experiments on three real-world TKGs demonstrate the superiority of MetaTKGR over state-of-the-art baselines by a large margin.

1 Introduction

Temporal Knowledge Graphs (TKGs) store temporally evolving facts (By fact, we refer to a subject entity has a relation with an object entity at some time, e.g., “Macron” was “elected” as “French president” in “2017”). It has been increasingly used in various knowledge-driven and time-sensitive applications, such as social event forecasting, question answering, and recommendation. Large scale TKGs are hard to obtain due to the time-varying nature of labels and the excessive cost of human annotation. As such, automating prediction and reasoning about missing facts over time has attracted more recent interest. However, most prior efforts are limited to reasoning about existing entities but neglecting the commonly observed low-data regime where many new entities emerge with extremely few observations as new events or topics evolve over time. This restricts their applicability.

Instead, inspired by the human ability to recognize new concepts from exposure to only very few instances, we propose and solve a practical and challenging few-shot temporal knowledge graph
reasoning problem that specifically aims to predict future facts for newly emerging entities with a few observed links on TKGs. Existing efforts [15][52][18] in similar settings simplify the task by ignoring the emergence of new entities over time and randomly simulating unseen entities on static knowledge graphs. This formulation, originating from few-shot learning in the vision domain [11][59], fails at capturing the real-world divergence between the distributions of seen and unseen entities. To the best of our knowledge, the generalized few-shot temporal reasoning task, that aims to imitate the fast learning ability of humans, has still not been formally investigated.

The challenges in solving this problem are two-fold: 1) **few-shot**: the extremely few facts associated with new entities cannot provide sufficient information for representation learning from scratch. 2) **time shift**: new entities usually exhibit different characteristics in the future, due to the time-evolving nature of TKGs. It leads models to generalize poorly in the future. Unfortunately, prior work fails to overcome these two challenges simultaneously.

On one hand, to enhance low-data entity characterization, a diverse range of neighborhood aggregation schemes that retrieve related information from other entities are explored, such as hard subgraph sampling methods (e.g., vanilla hop-based sampling [31], random sampling [16], personalized Pagerank sampling [66][44], importance sampling [6], etc) and soft sampling via trainable attention mechanisms [47][45][20]. However, those rigid methods cannot well adapt to the new entities with the time-varying distributions due to the lack of instant supervision. On the other hand, recent attempts [11][33][18][9] leverage meta-learning to improve unseen entity adaptation. While the setup for unseen entities is simulated by randomly splitting entity sets from static graphs, they fail to handle the more realistic temporal adaptation with large entity variances.

To overcome both the aforementioned challenges, we propose a novel Meta Temporal Knowledge Graph Reasoning (MetaTKGR) framework, where learning the strategies for sampling and aggregating neighbors from recent facts (to enhance the new entities’ predictions) can be dynamically adapted from signals of temporal supervision on their future facts as instant feedback. Intuitively, the global knowledge of sampling and aggregating can be extracted as learning to learn ability through such a meta-optimization, which takes the time-evolving nature of knowledge into consideration, avoiding progressive performance drift over time. Concretely, we formulate this learning strategy to be parameterized by a temporal encoder. On the simulated new entities during training phase, starting from the initial few-shot links, the temporal encoder can softly sample and attentively integrate relative and time-aware information from the temporal neighbors, making the strategy learning (neighbor sampling + aggregation) differentiable. The quantitative measurement of how well the current strategy performs can be reflected in the performance on predicting future facts allowing one to automatically adjust the strategy in turn. During the nested loop for meta-optimization, we firstly learn the temporal encoder on recent facts (inner loop), then gradually increase the difficulties by autoregressively adapting it to farther away future facts (outer loop). Furthermore, to better estimate the supervision signal given by future facts in different and unseen distributions, we adopt the PAC-Bayes method [45] to theoretically analyze the temporal adaptation bound on the future facts. This can serve as an adaptation regularizer to provide stability and improve the generalization ability of current strategy over time. Empirically, extensive experiments on three real-world TKGs validate the effectiveness of MetaTKGR, which significantly outperforms all baselines from static/temporal KG reasoning and few-shot (knowledge) graph learning areas, by up to 11.4% relative gain on average with competitive efficiency. To sum up, our main contributions are three-fold:

- **Problem formulation**: We explore a practical few-shot temporal knowledge graph reasoning setting, minimizing the gap with the realistic few-shot learning scenarios for humans;
- **Novel framework**: We propose a novel meta temporal reasoning framework on TKGs, namely MetaTKGR, to address both few-shot and time shift challenges;
- **Extensive evaluation**: the proposed MetaTKGR method demonstrates significant gains in effectiveness on three real-world TKGs over a diverse range of state-of-the-art baselines.

## 2 Problem Definition

In this section, we formally define the few-shot temporal knowledge graph reasoning task. First of all, a temporal knowledge graph can be defined as follows:

**Definition 2.1 (Temporal Knowledge Graph).** A temporal knowledge graph can be denoted as $G^T = \{ (e_o, r, e_o, t) \} \subseteq \mathcal{E}^T \times \mathcal{R} \times \mathcal{E}^T \times \mathcal{T}$, where $\mathcal{E}^T$ denotes a set of entities that appear in time
During the meta-training phase, we simulate a set of new entities from existing entity set by assuming observable facts of \( \{ \text{new entity} \} \) over time, even for a long time interval \( (T, T') \). Knowledge graph reasoning (KGR) is essentially the problem of predicting missing facts in the partially observed KG. While existing few-shot KGR models simulate unseen entities by randomly selecting from the existing entities, we focus specifically on predictions for newly emerging entities over time, which can be formalized as few-shot temporal knowledge graph reasoning task:

**Definition 2.3 (Few-shot Temporal Knowledge Graph Reasoning).** Given a temporal knowledge graph \( G^T \) collected no later than \( T \), for each new entity \( \tilde{e} \in E^\tilde{T} \setminus E^T \) (\( \tilde{T} > T \)), we assume the first \( K \) associated facts are observed: \( \{(\tilde{e}, r_i, e_i, t_i) \text{ or } (e_i, r_i, \tilde{e}, t_i)\}^{K}_{i=1} \). The task aims to predict the missing entities in the future facts \( (\tilde{e}, r, \tilde{e}, t) \) or \( (? , r, \tilde{e}, t) \) given the relation \( r \in R \) and specific time \( t \in (T, T') \). We further assume \( K \) is a small number, as a realistic setting.

3 Learning to Sample And Aggregate with MetaTKGR

In this section, we present a novel framework MetaTKGR to solve the few-shot temporal knowledge graph reasoning problem. We first introduce the basic setup of our learning objective and the overall framework, and then detail the temporal encoder module to represent new entities, followed by introduction of meta temporal reasoning for model training.

3.1 Learning Objective

Suppose we are given a temporal knowledge graph \( G^T = \{(s, r, e, t) \} \subseteq E^T \times R \times E^T \times T \) collected no later than \( T \). For each new entity \( \tilde{e} \) appearing within a later interval \( (T, T') \), we aim to predict future links \( (\tilde{e}, r, e, t) \) or \( (e, r, \tilde{e}, t) \) happening at timestamp \( t \in (T, T') \). Towards this goal, we measure correctness of each possible quadruple by a score function \( s(\cdot; \phi) \) parameterized by \( \phi \), and maximize the scores of true quadruples containing any new entities in order to rank them higher than all other false quadruples:

\[
\max_{\phi} E_{\tilde{e} \sim p(\tilde{E})}[s(\tilde{e}, r, e, t; \phi) \text{ or } s(e, r, \tilde{e}, t; \phi)], \quad \text{where } \tilde{E} = E^\tilde{T} \setminus E^T, \quad t \in (T, T'),
\]

where \( p(\tilde{E}) \) denotes distribution of all new entities, \( \phi \) denotes parameters of model to represent entity/relation into \( d \)-dimensional space \( h_e, h_r, h_e \in \mathbb{R}^d \). To represent \( \tilde{e} \) into \( h_e \) via \( \phi \), as aforementioned, the challenges lie in how to enable \( \phi \) to encode generalized knowledge (how to sample and aggregate temporal neighbors) that can be easily adapted to new \( \tilde{e} \) and achieve robust performance over time, even for a long time interval \( (T, T') \). We thereby formulate few-shot temporal knowledge graph reasoning as a meta-learning problem to extract such knowledge.

3.2 MetaTKGR Framework

Figure 4 shows the MetaTKGR framework. To be more formal, each task corresponds to each new entity \( \tilde{e} \) over distribution \( p(\tilde{E}) \), and the links can be represented as a chronological sequence \( \{(\tilde{e}, r_i, e_i, t_i) \text{ or } (e_i, r_i, \tilde{e}, t_i)\}_{i=1}^{N_{\tilde{E}}} \), where \( N_{\tilde{E}} \) denotes total number of links associated to \( \tilde{e} \). Then we divide them into support set \( S_{\tilde{E}} = \{(\tilde{e}, r_i, e_i, t_i) \text{ or } (e_i, r_i, \tilde{e}, t_i)\}_{i=1}^{N_{\tilde{E}}} \) and query set \( Q_{\tilde{E}} = \{(\tilde{e}, r_i, e_i, t_i) \text{ or } (e_i, r_i, \tilde{e}, t_i)\}_{i=K+1}^{N_{\tilde{E}}} \), where \( K \) denotes the amount of initially observable facts of \( \tilde{e} \).

During the meta-training phase, we simulate a set of new entities from existing entity set by assuming there are only few-shot links of these entities (support set \( S_{\tilde{E}} \)). The model first fine-tunes parameters \( \phi \) on \( S_{\tilde{E}} \) (inner loop), and then minimizes the predictive loss on corresponding query set \( Q_{\tilde{E}} \) using the updated parameters \( \phi_{\tilde{E}} \) (outer loop). It optimizes the global sampling and aggregation strategy as
which takes neighbor feature, relation
where
To better represent few-shot entities, a wider range of neighbors should be considered, as the close
propagated from them to the target entity. Also, the size of neighbors grows exponentially with
neighbors can dominate the aggregation process, as the information of multi-hop neighbors is all
In few-shot cases, such layer-by-layer procedure faces difficulties. For one thing, the limited one-hop
neural networks (GNNs)-based layers can integrate information from multi-hop neighbors [41][42].

Also, the size of neighbors grows exponentially with

temporal pattern of
evolving over time. The temporal en-
time neighbors from TKGs, then at-
tent. Towards this goal,
caused by few-shot links to some ex-
resolution ability over time (challenge 2), as they assume identical distributions between $S^e$ and $Q^e$, contradicting the facts that new entities are highly evolving over time on TKGs. To coherently
resolve the two challenges, we advance the existing few-shot KG reasoning works by proposing:

• **Temporal encoder**, which learns the time-aware representation of each new entity $\tilde{e}$ by
sampling and aggregating information from TKG neighbors on continuous domain.
• **Meta temporal reasoning**, which learns an optimal sampling and aggregating parameters via
bi-level optimization (inner optimization and outer optimization). The learned parameters can be easily updated to new entities and maintain temporal robustness.

### 3.3 Temporal Encoder

On temporal knowledge graphs, entities are evolving over time. The temporal en-
coder $f_{\phi}$ is to embed each entity $\tilde{e}$ into
low-dimensional latent space at each time: $h_{\phi}(t) \in \mathbb{R}^d$, where it can model the
temporal pattern of $\tilde{e}$ along time. By
doing so, it resolves the scarcity issues
caused by few-shot links to some ex-
tent. Towards this goal, $f_{\phi}$ first samples
temporal neighbors from TKGs, then at-
tentively aggregates information from
the temporal neighbors of each entity,
which takes neighbor feature, relation feature and time feature into account.

To better represent few-shot entities, a wider range of neighbors should be considered, as the close
neighbors around the new entities are usually insufficient. Conventionally, stacking several graph
neural networks (GNNs)-based layers can integrate information from multi-hop neighbors [41][42].
In few-shot cases, such layer-by-layer procedure faces difficulties. For one thing, the limited one-hop
neighbors can dominate the aggregation process, as the information of multi-hop neighbors is all
propagated from them to the target entity. Also, the size of neighbors grows exponentially with
the increase of GNN layers, as they collect all neighbors in each hop without strategic selection,
causing severe efficiency issues. Instead, our temporal encoder first samples multi-hop neighbors via a time-bounded breadth-first-search algorithm, then aggregates information directly from the sampled neighbors in an attentive manner. As summarized in Algorithm 1 at each time, it selectively samples up to $b$ multi-hop temporal neighbors $\mathcal{N}_e(t)$ which have interactions within a recent time range $\Delta t$. Given the temporal neighbor $\mathcal{N}_e(t)$ for each new entity, temporal encoder $f_\phi$ represents a new entity as $h_e(t)$ at time $t$:

$$h_e^l(t) = \sigma \left( \sum_{(e, r, e', t) \in \mathcal{N}_e(t)} \alpha_{e, e'} \left( h_{e'}^{l-1}(t) W \right) \right),$$

where $l$ denotes the layer number, $\sigma(\cdot)$ denotes the activation function $ReLU$, $\alpha_{e, e'}$ denotes the attention weight of entity $e$ to new entity $e'$, and $W$ is the trainable transformation matrix. To aggregate from history, $\alpha_{e, e'}$ is supposed to be aware of entity feature, time delay and topology feature induced by relations. Thus, we design $\alpha_{e, e'}$ as follows:

$$\alpha_{e, e'} = \frac{\exp(q_{e, e'})}{\sum_{(e, r, e', t) \in \mathcal{N}_e(t)} \exp(q_{e, e'})}, \quad q_{e, e'} = a \left( \| h_{e'}^{l-1} \| \| h_e^{l-1} \| \| \Phi(t - t_i) \| \right),$$

where $q_{e, e'}$ measures the pairwise importance by considering the entity embedding, relation embedding and time embedding, $a \in \mathbb{R}^{4d}$ is the shared parameter in the attention mechanism. Following §2 we adopt random Fourier features as time encoding $\Phi(\Delta t)$ to reflect the time difference.

### 3.4 Meta Temporal Reasoning

Let $h_e^l(t)$ denote the representations of new entity $e$ produced by $f_\phi$ with $L$ layers. For each quadruple $(e, r, e', t)$, we employ a translation-based score function $f_\phi$ to measure the correctness:

$$s(e, r, e', t) = -\| h_e^l(t) + h_e - h_e^l(t) \|^2.$$  

We first fine-tune $f_\phi$ on support set $\mathcal{S}_e$ for entity-specific parameters $f_{\phi_0}$, and then optimize training loss on query set $\mathcal{Q}_e$ to learn the global parameters shared by all entities. To calculate the predictive loss, since each set only contains positive quadruples, we perform negative sampling $\mathcal{Q}_e$ to update MetaTKGR by training it to rank positive quadruples higher than negative ones. Specifically, taking the support set $\mathcal{S}_e$ of entity $e$ as an example, we construct a negative set $\mathcal{S}_e^- = \{(e, r, e^-, t) \text{ or } (e^-, r, e^-, t)\}$ by replace the ground truth entity $e$ with a corrupted entity $e^-$. We then use empirical hinge loss on the given set as follows:

$$\hat{L}(f_{\phi_0}, \mathcal{S}_e) = \sum_{(e, r, e, t) \in \mathcal{S}_e} \sum_{(e, r, e^-, t) \in \mathcal{S}_e^-} \max \left( \gamma - s(e, r, e, t) + s(e, r, e^-, t), 0 \right),$$

where $\gamma > 0$ is a margin value to distinguish positive and negative quadruples.

**Inner optimization:** For each task corresponding to new entity $e \sim p(\mathcal{E})$, we first adapt the global parameter $\phi$ for each new entity $e$ by minimizing the predictive loss on the support set $\mathcal{S}_e$:

$$\phi_e = \phi - \eta \frac{\partial \hat{L}(f_{\phi}, \mathcal{S}_e)}{\partial \phi},$$

where $\eta$ is the inner loop learning rate.

By Eq. 6 we simulate the adaption for new entities. Updating from an optimal global parameter $\phi$, the model is fine-tuned into entity-specific for new entities. Thus, it is crucial to design a meta-learning strategy to learn an optimal global parameter $\phi$ with a robust generalization ability over time.

**Outer optimization:** Since new entities evolve over time, it is not only $Q_e$ but also each part of $Q_e$ that follows different distributions. Thus, instead of optimizing the meta-learner on the whole query set $Q_e$ at

**Algorithm 2: MetaTKGR: Meta-training.**

**Input:** Temporal knowledge graph $G^T \subseteq \mathcal{E}^T \times \mathcal{R} \times \mathcal{E}^T \times \mathcal{T}$, randomly initialized $\phi$.

**Output:** Learned global parameter $\phi$.

1. Simulate new entity set $E$ from $\mathcal{E}^T$;
2. Train model parameter $\phi$ from many-shot entities $\mathcal{E}^T \setminus E$;
3. Construct $\mathcal{S}_e$ and $Q_e$ for each new entity $e$, where $Q_e = \{Q_e^{(1)}, Q_e^{(2)}, \cdots, Q_e^{(M)}\}$;
4. while $\phi$ not converge do
   1. for each time interval $m$ do
      1. for each new entity $e$ do
         1. for each new entity $e$ do
            1. Compute $\hat{L}(f_{\phi_e}, \mathcal{S}_e)$ on $\mathcal{S}_e$ by Eq. 5;
            1. Perform adaptation by Eq. 6 to update $\phi_e$;
            1. Calculate $\hat{L}(f_{\phi_e}, Q_e^{(m)})$ by Eq. 5;
            1. Calculate temporal adaptation regularizer by Eq. 8;
            1. Update $\phi^{(m)}$ by Eq. 9;
      1. End
   1. End
   1. Update $\phi$ by Eq. 9;
   1. End
5. End
where we first adapt entity-specific parameters \( \phi_\varepsilon \) to recent facts, then gradually increase the difficulties by autoregressively adapting it to farther away future facts. Through this process, we are able to guide and stabilize the training of global parameters via a temporal signal. Specifically, we split the time span of query sets into \( M \) time intervals, where each new entity has a sequence of query set corresponding to different time intervals \( Q_\varepsilon = \{ Q_\varepsilon^{(1)}, Q_\varepsilon^{(2)}, \ldots, Q_\varepsilon^{(M)} \} \). We first adapt and optimize \( \phi_\varepsilon \) for each new entity on \( Q_\varepsilon^{(1)} \) for \( \phi^{(1)} \) encoding global knowledge for predictions in the 1-st time interval. Then we gradually adapt each \( \phi_\varepsilon \) fine-tuned from last time interval on farther away intervals until \( \phi^{(M)} \) is learned, which maintain temporal robustness from time interval 1 to \( M \).

We then discuss how to measure the feedback (predictive loss) from each adaptation. To simulate real scenarios, we assume query sets to follow different and unseen distributions. Taking adaptation from \( m \)-th time interval to \( m + 1 \)-th time interval as example, we view \( p(\phi^{(m)}) \) as prior parameter distribution and aim to learn the posterior parameter distribution \( q(\phi^{(m+1)}) \) conditioning on query set in \( m + 1 \)-th interval. The unseen query set distribution in \( m + 1 \)-th interval prevent us to estimate \( q(\phi^{(m+1)}) \) by either using unbiased empirical loss or Bayes rules. To resolve this, we instead adopt the PAC-Bayes method \([35]\), which allows one to learn a parameter posterior that fits the observations without knowing the observations distribution. We propose the following theorem to relate real prediction loss in new time interval with its empirical version:

**Theorem 3.1 (PAC-Bayes Generalization Bound on Temporal Adaptation).** Let \( D^{(m+1)} = \bigcup_{\varepsilon \in \mathcal{E}} Q_\varepsilon^{(m+1)} \) denote all query sets in \( m + 1 \)-th time interval of all new entities from \( \mathcal{E} \). For any \( \delta \in (0, 1) \) and learned parameter distribution \( p(\phi^{(m)}) \) in last time interval, with probability at least \( 1 - \delta \) on \( D^{(m+1)} \):

\[
L(f_\phi^{(m)}, D^{(m+1)}) \leq \hat{L}(f_\phi^{(m)}, D^{(m+1)}) + \sqrt{\frac{\text{KL}(q(\phi^{(m+1)})||p(\phi^{(m)})) + \log \frac{|D^{(m+1)}|}{\delta}}{2|D^{(m+1)}| - 1}},
\]

where \( L(f_\phi^{(m)}, D^{(m+1)}) = \mathbb{E}_{\varepsilon \sim p(\mathcal{E})} \left[ L(f_\phi^{(m)}, Q_\varepsilon^{(m+1)}) \right] \) denotes real predictive loss on new time interval with new and unknown data distribution, \( \hat{L}(f_\phi^{(m)}, D^{(m+1)}) \) denote the empirical version of the loss, and \( \text{KL}(||\cdot||) \) is the Kullback-Leibler divergence.

Readers can refer to Appendix A.1 for proof. Thus, using Theorem 3.1, we can adapt \( \phi^{(m)} \) to \( m + 1 \) time interval and estimate the feedback (predictive loss) for our meta-learning framework via the following temporal adaptation regularizer:

\[
\mathcal{R}(f_\phi^{(m)}; D^{(m+1)}) = \hat{L}(f_\phi^{(m)}, D^{(m+1)}) + \sqrt{\frac{\text{KL}(q(\phi^{(m+1)})||p(\phi^{(m)})) + \log \frac{|D^{(m+1)}|}{\delta}}{2|D^{(m+1)}| - 1}}.
\]

**Remark.** To interpret the temporal adaptation regularizer \( \mathcal{R}(f_\phi^{(m)}; D^{(m+1)}) \), it can be viewed as a combination of empirical risk on the query set in the new time interval as well as a regularizer of parameter distribution across time intervals in form of KL-divergence. The regularizer along time domain can guarantee that global knowledge \( \phi \) is trained by predictive losses from all time intervals without overfitting in specific time interval. Thus, we can improve the generalization ability of our meta-learner over time by the following update step by step, from \( m = 1 \) to \( m = M \):

\[
\phi^{(m+1)} = \phi^{(m)} - \beta \frac{\partial \mathcal{R}(f_\phi^{(m)}; D^{(m+1)})}{\partial \phi},
\]

where \( \phi^{(0)} \) can be obtained by training on existing entities. The meta-training phase of MetaTKGR is summarized in Algorithm 2. During meta-test phase, given a new entity \( \hat{e} \), we can first fine-tune \( \phi \) on few-shot links, and then utilize \( f_{\phi_{\hat{e}}} \) to represent \( \hat{e} \) for future predictions.

4 **Experiments**

**Datasets.** We evaluate the proposed MetaTKGR framework on three public TKGs, where YAGO \([34]\) and WIKI \([22]\) stores time-varying facts and ICEWS18 \([3]\) is event-centric. They are collected from different time ranges and have different time units, which can validate our framework in the context of various types of evolution. Notably, most new entities appear with few-shot facts, validating the practical value of our proposed setting. Table 1 shows the detailed statistics. We report detailed dataset descriptions as well as entity links distribution in the Appendix A.2.
with a ratio of 0.4:0.25:0.1:0.25 chronologically, then we collect the entities that firstly appear in T-GEN
Table 2: The results of 3-shot temporal knowledge graph reasoning. Average results on LAN, I-GEN, T-GEN
future predictions. Compared with MetaDyGNN, that does not optimize for new entities, as they ignore the temporal information and perform poorly in all datasets, although they are trained on both existing entities and new entities with few-shot links.
Baselines (demonstrating the superiority of respectivelly. On average, it is worth noting that we measure the ranks in a filtered setting, where we filter other true quadruples, and adopt mean reciprocal rank (MRR) scheme to evaluate the performance. Specifically, we rank all entities at the missing position in Evaluation Protocol and Metrics.
Experimental Setup. Given the temporal knowledge graph, we first split the time duration into four
periods as background/meta-training/meta-validation/meta-test entity set. We train our model
in Appendix A.3. Framework for few-shot link prediction on homogeneous graphs. We describe the baselines in detail in Appendix A.4.
<table>
<thead>
<tr>
<th>Baselines</th>
<th>1-shot</th>
<th>2-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>0.183</td>
<td>0.215</td>
</tr>
<tr>
<td>TransR</td>
<td>0.198</td>
<td>0.218</td>
</tr>
<tr>
<td>RotatE</td>
<td>0.160</td>
<td>0.196</td>
</tr>
<tr>
<td>RE-NET</td>
<td>0.233</td>
<td>0.248</td>
</tr>
<tr>
<td>RE-GCN</td>
<td>0.175</td>
<td>0.203</td>
</tr>
<tr>
<td>LAN</td>
<td>0.174</td>
<td>0.181</td>
</tr>
<tr>
<td>I-GEN</td>
<td>0.202</td>
<td>0.248</td>
</tr>
<tr>
<td>T-GEN</td>
<td>0.206</td>
<td>0.237</td>
</tr>
<tr>
<td>MetaDyGNN</td>
<td>0.241</td>
<td>0.265</td>
</tr>
<tr>
<td>MetaTKGR</td>
<td>0.249</td>
<td>0.262</td>
</tr>
<tr>
<td>Gains (%)</td>
<td>5.68</td>
<td>12.21</td>
</tr>
<tr>
<td></td>
<td>9.00</td>
<td>9.19</td>
</tr>
<tr>
<td></td>
<td>6.26</td>
<td>6.38</td>
</tr>
<tr>
<td></td>
<td>8.47</td>
<td>6.35</td>
</tr>
<tr>
<td></td>
<td>9.11</td>
<td>14.85</td>
</tr>
<tr>
<td></td>
<td>8.99</td>
<td>12.4</td>
</tr>
</tbody>
</table>

Table 3: The results of 1-shot and 2-shot experiment. We report complete results in Appendix A.5.

Baseline. We compare nine state-of-the-art baselines from four related areas: 1) TransE, 2) TransR, 3) RotatE, 4) RE-NET, 5) RE-GCN, 6) Temporal knowledge graph embedding methods; 7) LAN, 8) I-GEN, 9) T-GEN; 10) Few-shot methods on static knowledge graph; 11) MetaDyGNN. A meta-learning framework for few-shot link prediction on homogeneous graphs. We describe the baselines in detail in Appendix A.3.

Experimental Setup. Given the temporal knowledge graph, we first split the time duration into four with a ratio of 0.4:0.25:0.1:0.25 chronologically, then we collect the entities that firstly appear in each period as background/meta-training/meta-validation/meta-test entity set. We train our model on background set as initialization and simulate few-shot tasks on meta-training set. To construct few-shot tasks, we assume the first K = 1, 2, 3 quadruples of new entities are known and report the performance of the remaining quadruples in the future. For fair comparison, we keep the dimension of all embeddings as 128, and utilize pre-trained 1-shot TransE embeddings for initialization for models if applicable. We report detailed experimental setup, especially of MetaTKGR, in the Appendix A.4.

Evaluation Protocol and Metrics. For each prediction (e, r, ?, t) or (?; r, e, t), we use ranking scheme to evaluate the performance. Specifically, we rank all entities at the missing position in quadruples, and adopt mean reciprocal rank (MRR) and Hits at 1, 3, 10 (H@1,3,10) as evaluation metrics. It is worth noting that we measure the ranks in a filtered setting, where we filter other true quadruples existing in datasets, following [33, 46].

Main Results. Table 2 and Table 3 report overall result for K = 3 and K = 1, 2 few-shot experiments respectively. On average, MetaTKGR achieves 11.4% relative improvement over best baseline model, demonstrating the superioritly of MetaTKGR in modeling new entity evolution in low-data regime. Baselines (TransE, TransR, RotatE, RE-NET) that do not optimize for new entities perform poorly on all datasets, although they are trained on both existing entities and new entities with few-shot links. LAN, I-GEN, T-GEN also produce unsatisfying results, despite the fact that they propose special designs to represent new entities. Note that they have worse performance in some cases than RE-NET that does not optimize for new entities, as they ignore the temporal information and perform poorly in future predictions. Compared with MetaDyGNN, MetaTKGR shows consistently better performance,
because our temporal encoder can handle multi-relational graphs in low-data regimes better, and our
temporal meta-learning framework can produce more robust predictions.

**Performance of Prediction over Time.** Next,
we study the performance of MetaTKGR over time.
Figure 2 shows the performance comparisons of
3-shot predictions over different timestamps on
the YAGO and ICEWS18
datasets, measured by
filtered $H@10$ metrics. MetaTKGR consistently beats all strong baselines. Although the compared
baselines can optimize newly emerging entities, they perform poorly in wide time intervals because
they largely ignore the temporal information and the distribution discrepancy caused by evolution.
We notice that the relative gains of our model get more significant with increasing time steps. It
illustrates that our temporal meta-learning framework can improve the generalization ability over
time. Performance on ICEWS18 fluctuates more severely. This is expected since the evolution of
ICEWS18 is not stable due to the much shorter time unit (day).

**Ablation Study.** We evaluate performance improvements brought by the temporal meta-learning
framework by following ablations: 1) **Fine-Tuning** is trained on existing entities without using any
meta-learning strategy, and then fine-tuned on new entities; 2) **Static Meta-Learning** utilizes conven-
tional MAML to train models; 3) **MetaTKGR w/o Regularizer** meta-trains model parameters on each
time interval step by step, without explicitly optimizing the temporal adaptation regularizer. Figure 3
reports the results measured by filtered $H@10$. We can conclude that modeling the distribution
discrepancy caused by the temporal evolution and optimizing the temporal adaptation regularizer can
improve the performance in the future prediction. Also, simply fine-tuning the model parameters on
new entities leads to poor results, as during background phase, the model does not extract generalized
knowledge that can be easily adapted to new entities.

![Figure 2: Performance of YAGO (left) and ICEWS18 (right) over time.](image1)

![Figure 3: Ablation Studies, evaluated by filtered $Hit@10$.](image2)

![Figure 4: MRR over training time.](image3)

### Table 4: Cross-shot results on YAGO.

<table>
<thead>
<tr>
<th>Test</th>
<th>1-Shot (Training)</th>
<th>3-Shot (Training)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>$H@1$</td>
</tr>
<tr>
<td>1-S</td>
<td>0.294</td>
<td>0.240</td>
</tr>
<tr>
<td>3-S</td>
<td>0.354</td>
<td>0.297</td>
</tr>
<tr>
<td>5-S</td>
<td>0.377</td>
<td>0.360</td>
</tr>
<tr>
<td>R-S</td>
<td>0.358</td>
<td>0.304</td>
</tr>
</tbody>
</table>

**Cross-shot Learning.** In reality, new entities are usually associated with various numbers of facts
initially. Thus, the robustness of models on different numbers of shots during testing phase is critical.
To simulate such scenario, we evaluate MetaTKGR by varying the number of shots including 1-, 3-, 5-, and random-shot (R-S: between 1 and 5) during meta-training and meta-test phases. Table 4 reports
the cross-shot learning results. As expected, with the increase of observable shots during testing
phase, the performance becomes better. The differences in the number of shots used for training do
not significantly affect the results, demonstrating that MetaTKGR trained with a fixed number of shots
performs robustly under the various number of shots during testing.
We study a realistic but underexplored few-shot temporal knowledge graph reasoning problem, which aims at predicting future facts for newly emerging entities with a few facts. To this end, we propose a novel Meta Temporal Knowledge Graph Reasoning framework MetaTKGR. It meta-learns the global knowledge of sampling and aggregating temporal neighbors, which can be adapted quickly to new entities for future prediction. Such procedure is gradually guided by the performance on predicting future facts, from near time intervals to far away ones. We further theoretically analyze and propose a temporal adaptation regularizer to stabilize and generalize the learned knowledge on future tasks. We empirically validate the effectiveness of MetaTKGR on three real-world temporal knowledge graphs, on which the proposed framework significantly outperforms an extensive set of SOTA baselines.

5 Related Work

Few-shot Learning on Knowledge Graph. To alleviate data scarcity, various advanced techniques have emerged, such as transfer learning [31, 55], semi-supervised learning [5], domain adaptation [30, 27, 28, 26] and few-shot learning (FSL) [56, 67]. Few-shot learning aims at learning generalized knowledge from existing tasks to extract transferable priors for new tasks with few labeled samples, including metric learning based approaches [50, 55] and meta-learning based approaches [11, 39, 12, 25]. For knowledge graph reasoning, the success of few-shot learning has facilitated learning the representations of scarce entities [15, 52, 43, 11] and scarce relations [68, 58, 47, 56] respectively. In this paper, we are primarily interested in works to represent entities with few-shot facts, as relation set is relatively stable along time [44]. Given a new entity linked with few-shot facts, the neighborhood aggregator used in graph neural networks (GNNs) reduces the inductive bias via structural knowledge [21, 16, 51]. KL computes the representations of few-shot entities by GNN-based neighboring aggregation scheme. [52, 43, 56] further extend it by utilizing attention mechanisms. However, those models are usually trained on both many-shot and few-shot entities, ending up being suboptimal for few-shot entities. To address it, a recent work [11] explores how to mate-train the neighborhood aggregator so as to effectively adapt global knowledge for few-shot knowledge reasoning. Deep graph learning has been attracting enormous interest recently [21, 16, 51, 61, 63, 62, 60, 23]. More broadly on graphs, Meta-Graph [4], G-Meta [18] and MetaDyGNN [65] are proposed to utilize meta-learning for link prediction across multiple graphs, on static/temporal homogeneous graphs. However, most works are designed for few-shot learning on static (knowledge) graphs. How to aggregate neighbors considering time factors and optimize for long-term performance are unsolved. MetaDyGNN [65] is a recent framework combining MAML with TGAT [2] on temporal homogeneous graph. But it did not consider distribution difference between few-shot facts and future facts caused by the evolution of new entities either, leading to increasingly poor performance over time, especially on temporal knowledge graphs with more complicated structures.

Temporal Knowledge Graph Reasoning. Temporal knowledge graphs (TKGs) store time-varying facts in the real-world. Temporal knowledge graph reasoning aims to predict missing facts at a certain time in the future. It is mostly formulated as measuring the correctness of factual samples and negative samples by specially designed score functions [2, 33, 39, 46, 13]. Compared with static KG reasoning tasks [13], the main challenge lies in how to incorporate time information into the representation process. Several embedding-based methods have been proposed. They encode time-dependent information of entities and relations by decoupling embeddings into static component and time-varying component [59, 14], utilizing recurrent neural networks (RNNs) to adaptively learn the dynamic evolution from historical fact sequence [20, 47], or learning a sequence of evolving representations from discrete knowledge graph snapshots [47, 32, 20]. However, all of the existing temporal KG reasoning models aim to extrapolate future facts among existing entities, and how to predict future facts specifically for new emerging entities is largely under-explored.

6 Conclusion

We study a realistic but underexplored few-shot temporal knowledge graph reasoning problem, which aims at predicting future facts for newly emerging entities with a few facts. To this end, we propose a novel Meta Temporal Knowledge Graph Reasoning framework MetaTKGR. It meta-learns the global knowledge of sampling and aggregating temporal neighbors, which can be adapted quickly to new entities for future prediction. Such procedure is gradually guided by the performance on predicting future facts, from near time intervals to far away ones. We further theoretically analyze and propose a temporal adaptation regularizer to stabilize and generalize the learned knowledge on future tasks. We empirically validate the effectiveness of MetaTKGR on three real-world temporal knowledge graphs, on which the proposed framework significantly outperforms an extensive set of SOTA baselines.
Acknowledgments and Disclosure of Funding

The authors would like to thank Tianshi Wang, Yuchen Yan, Jialu Wang for their helpful comments and discussion. The authors would also like to thank the anonymous reviewers of NeurIPS for valuable comments and suggestions. Research reported in this paper was sponsored in part by DARPA award HR001121C0165, DARPA award HR00112290105, Basic Research Office award HQ00342110002, and the Army Research Laboratory under Cooperative Agreement W911NF-17-20196.

References


[69] Yuyue Zhao, Xiang Wang, Jiawei Chen, Yashen Wang, Wei Tang, Xiangnan He, and Haiyong Xie. Time-aware path reasoning on knowledge graph for recommendation. ACM Trans. Inf. Syst., 2022.