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

Anonymous Author(s)
Affiliation
Address
email

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 eight state-of-the-art baselines by a large margin.

1 Introduction

Temporal knowledge graphs (TKGs) [28, 3, 22] store temporally evolving facts (By facts, 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 [32, 23], question answering [35], and recommendation [57, 56]. 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 [42, 9, 12, 43, 21]. 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 [39]. 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. For example, as shown in Figure 1, there is only one observable fact about a new person joining the TKG, yet our goal is to predict their missing facts in the future. Existing efforts [16, 46, 1, 7] in similar settings simplify the task by ignoring the emergence of

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

new entities over time and randomly simulating unseen entities on static knowledge graphs. This formulation, originating from few-shot learning in the vision domain [10, 34], 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.

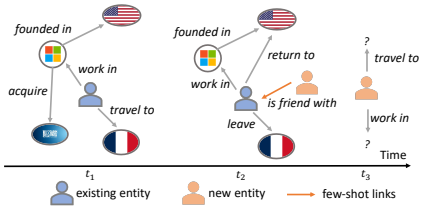

Figure 1: An illustrative example of 1-shot reasoning task over temporal KG.

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 [36], random sampling [17], personalized Pagerank sampling [54], importance sampling [5], etc) and soft sampling via trainable attention mechanisms [42, 43, 21]. 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 [1, 55, 20, 4] 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 [29] 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* $\mathcal{G}^T = \{(e_s, r, e_o, t)\} \subseteq \mathcal{E}^T \times \mathcal{R} \times \mathcal{E}^T \times \mathcal{T}$*, where* $\mathcal{E}^T$ *denotes a set of entities that appear in time interval* $(0, T)$*,* $\mathcal{R}$ *denotes relation set, and* $\mathcal{T}$ *denotes timestamp set. Each temporal link* (e_s, r, e_o, t) *refers to the fact that a subject entity* $e_s \in \mathcal{E}^T$ *has a relation* $r \in \mathcal{R}$ *with an object entity* $e_o \in \mathcal{E}^T$ *at timestamp* $t \in \mathcal{T}$*.*

As a temporal knowledge graph evolves, new entities continuously emerge, leading to an expansion of the entities set $\mathcal{E}^T$ over time. We further define new entities as follows:

Definition 2.2 (**New Entities in a Temporal Knowledge Graph**). *Given a time interval* (T, T') *(*$T' > T$*), entities that join the graph during* (T, T')*, i.e.,* $\tilde{e} \in \mathcal{E}^{T'} \setminus \mathcal{E}^T$ *(*$\setminus$ *denotes setminus), are defined as new entities on 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* $\mathcal{G}^T$ *collected no later than* T*, for each new entity* $\tilde{e} \in \mathcal{E}^{T'} \setminus \mathcal{E}^T (T' > T)$*, we assume the first* K *associated facts are observed:* $\{(\tilde{e}, r_i, e_i, t_i)$ *or* $(e_i, r_i, \tilde{e}, t_i)\}_{i=1}^K$*. The task aims to predict the missing entities in the future facts* $(\tilde{e}, r, ?, t)$ *or* $(?, r, \tilde{e}, t)$ *given the relation* $r \in \mathcal{R}$ *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 $\mathcal{G}^T = \{(e_s, r, e_o, t)\} \subseteq \mathcal{E}^T \times \mathcal{R} \times \mathcal{E}^T \times \mathcal{T}$ collected no later than time 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 ϕ, and maximize the scores of true quadruples containing any new entities in order to rank them higher than all other false quadruples:

$$\max_{\phi} \mathbb{E}_{\tilde{e} \sim p(\tilde{\mathcal{E}})}[s(\tilde{e}, r, e, t; \phi) \text{ or } s(e, r, \tilde{e}, t; \phi)], \text{ where } \tilde{\mathcal{E}} = \mathcal{E}^{T'} \setminus \mathcal{E}^T,\ t \in (T, T'), \tag{1}$$

where $p(\tilde{\mathcal{E}})$ denotes distribution of all new entities, ϕ denotes parameters of model to represent entity/relation into d-dimensional space $\mathbf{h}_{\tilde{e}}, \mathbf{h}_e, \mathbf{h}_r \in \mathbb{R}^d$. To represent $\tilde{e}$ into $\mathbf{h}_{\tilde{e}}$ via ϕ , as aforementioned, the challenges lie in how to enable ϕ 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 2 shows the MetaTKGR framework. To be more formal, each task corresponds to each new entity $\tilde{e}$ over distribution $p(\tilde{\mathcal{E}})$, and the links can be represented as a chronological sequence $\{(\tilde{e}, r_i, e_i, t_i)$ or $(e_i, r_i, \tilde{e}, t_i) | t_i \in (T, T'), t_i \leq t_j$ if $i < j\}_{i=1}^{N_{\tilde{e}}}$, where $N_{\tilde{e}}$ denotes total number of links associated to $\tilde{e}$. Then we divide them into support set $\mathcal{S}_{\tilde{e}} = \{(\tilde{e}, r_i, e_i, t_i)$ or $(e_i, r_i, \tilde{e}, t_i)\}_{i=1}^K$, and query set $\mathcal{Q}_{\tilde{e}} = \{(\tilde{e}, r_i, e_i, t_i)$ 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 $\mathcal{S}_{\tilde{e}}$). The model first fine-tunes parameters ϕ on $\mathcal{S}_{\tilde{e}}$ (*inner loop*), and then minimizes the predictive loss on corresponding query set $\mathcal{Q}_{\tilde{e}}$ using the updated parameters $\phi_{\tilde{e}}$ (*outer loop*). It optimizes the global sampling and aggregation strategy as generalized knowledge (i.e., learning to learn ability) by this bi-level optimization, as this procedure mimics the normal machine learning and inference process. For simplicity, to represent new entity $\tilde{e}$ along time, we denote $f_{\phi_{\tilde{e}}}$ as the temporal encoder parameterized by $\phi_{\tilde{e}}$, $\mathcal{L}(f_{\phi_{\tilde{e}}}, \mathcal{S}_{\tilde{e}})$ as model

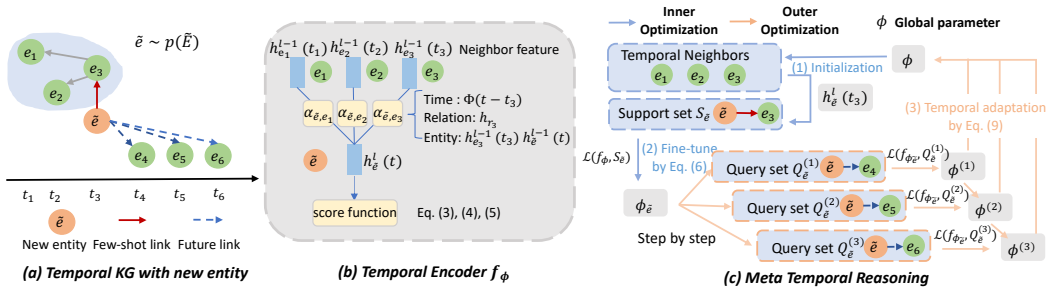

Figure 2: An overview of MetaTKGR framework. (a) Task illustration, defined in Definition 2.3; (b) Temporal encoder parameterized by ϕ samples and aggregates information from temporal neighbors, detailed in Section 3.3; (c) Meta temporal reasoning adopts a bi-level optimization to learn ϕ, detailed in Section 3.4: In the **inner optimization**: MetaTKGR initializes entity-specific parameters from global parameters via step (1), and fine-tunes them on support set via step (2) to optimize entity-specific parameters; In the **outer optimization**, MetaTKGR optimizes global parameters on query set via step (3) to learn good and robust global parameters. MetaTKGR alternatively updates model parameters via inner and outer optimization, in order to make global parameters be easily adapted to new entities and maintain temporal generalization ability. Through such bi-level optimization, MetaTKGR addresses the few-shot and time shift challenges.

predictive loss on $\mathcal{S}_{\tilde{e}}$. Model-agnostic meta-learning (MAML) [10] can be utilized to fulfill this goal as follows:

$$\phi^* \longleftarrow \arg\min_{\phi} \mathbb{E}_{\tilde{e} \sim p(\tilde{\mathcal{E}})} \left[\mathcal{L}(f_{\phi_{\tilde{e}}}, \mathcal{Q}_{\tilde{e}}) \right], \quad \text{where } \phi_{\tilde{e}} = \phi - \eta \frac{\partial \mathcal{L}(f_\phi, \mathcal{S}_{\tilde{e}})}{\partial \phi}, \tag{2}$$

where η denotes inner-loop learning rate. However, this training strategy cannot guarantee good generalization ability over time (*challenge 2*), as they assume identical distributions between $\mathcal{S}_{\tilde{e}}$ and $\mathcal{Q}_{\tilde{e}}$, 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 adapted to new entities and maintain temporal robustness.

3.3 Temporal Encoder. On temporal knowledge graphs, entities are evolving over time. The temporal encoder f_ϕ is designed to embed each entity $\tilde{e}$ into low-dimensional latent space at each time: $\mathbf{h}_{\tilde{e}}(t) \in \mathbb{R}^d$, 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 extent. Towards this goal, f_ϕ first samples temporal neighbors from TKGs, then attentively aggregates information from the temporal neighbors of each entity, which takes neighbor feature, relation feature and time feature into account.

Algorithm 1: Temporal Neighbor Sampler.

Input: New entity $\tilde{e}$, temporal knowledge graph $\mathcal{G}^t$, current timestamp t, neighbor budget b, time bound Δt.
Output: Temporal Neighbor $\mathcal{N}_{\tilde{e}}(t)$.
Initialize $Queue \leftarrow \tilde{e}$, $\mathcal{N}_{\tilde{e}}(t) \leftarrow \{\}$;
while $|\mathcal{N}_{\tilde{e}}(t)| < b$ *and* $Queue$ *is not empty* **do**
  $e \leftarrow Queue.dequeue()$;
  for *each* (e_s, r, e_o, t) *in* $\mathcal{G}^t$ **do**
    if $e_s == e$ *and* $t > t_{max} - \Delta t$ **then**
      Add (e_s, r, t) to $\mathcal{N}_{\tilde{e}}(t)$, Add e_s to $Queue$;
    if $e_o == e$ *and* $t > t_{max} - \Delta t$ **then**
      Add (e_o, r, t) to $\mathcal{N}_{\tilde{e}}(t)$, Add e_o to $Queue$;

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 [36, 17]. 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 $\mathcal{N}_{\tilde{e}}(t)$ which have interactions within a recent time range Δt. Given the temporal neighbor $\mathcal{N}_{\tilde{e}}(t)$ for each new entity, temporal encoder f_ϕ represents a new entity as $\mathbf{h}_{\tilde{e}}(t)$ at time t:

$$\mathbf{h}_{\tilde{e}}^{l}(t) = \sigma\left(\sum_{(e_i,r_i,t_i)\in\mathcal{N}_{\tilde{e}}(t)} \alpha_{\tilde{e},e_i}\left(\mathbf{h}_{e_i}^{l-1}(t_i)\mathbf{W}\right)\right), \tag{3}$$

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

$$\alpha_{\tilde{e},e_i} = \frac{\exp(q_{\tilde{e},e_i})}{\sum_{(e_k,r_k,t_k)\in\mathcal{N}_{\tilde{e}}(t)}\exp(q_{\tilde{e},e_k})}, \quad q_{\tilde{e},e_i} = \mathbf{a}\left(\mathbf{h}_{\tilde{e}}^{l-1}\|\mathbf{h}_{e_i}^{l-1}\|\mathbf{h}_{r_i}\|\Phi(t-t_i)\right), \tag{4}$$

where $q_{\tilde{e},e_i}$ measures the pairwise importance by considering the entity embedding, relation embedding and time embedding, $\mathbf{a}\in\mathbb{R}^{4d}$ is the shared parameter in the attention mechanism. Following [8] we adopt random Fourier features as time encoding $\Phi(\Delta t)$ to reflect the time difference.

3.4 Meta Temporal Reasoning. Let $\mathbf{h}_{\tilde{e}}^{L}(t)$ denote the representations of new entity $\tilde{e}$ produced by f_ϕ with L layers. For each quadruple $(\tilde{e},r,e,t)$, we employ translation-based score function [2] to measure the correctness: $s(\tilde{e},r,e,t;\phi) = -\|\mathbf{h}_{\tilde{e}}^{L}(t)+\mathbf{h}_r-\mathbf{h}_e^{L}(t)\|^2$. We first fine-tune f_ϕ on support set $\mathcal{S}_{\tilde{e}}$ for entity-specific parameters $f_{\phi_{\tilde{e}}}$, and then optimize training loss on query set $\mathcal{Q}_{\tilde{e}}$ 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 [2] to update MetaTKGR by training it to rank positive quadruples higher than negative ones. Specifically, taking the support set $\mathcal{S}_{\tilde{e}}$ of entity $\tilde{e}$ as an example, we construct a negative set $\mathcal{S}_{\tilde{e}}^{-} = \{(\tilde{e},r,e^{-},t) \text{ or } (e^{-},r,\tilde{e},t)\}$ by replace the ground truth entity e with a corrupted entity e^{-}. We then use empirical hinge loss on the given set as follows:

$$\hat{\mathcal{L}}(f_{\phi_{\tilde{e}}},\mathcal{S}_{\tilde{e}}) = \sum_{(e_s,r,e_o,t)\in\mathcal{S}_{\tilde{e}}}\ \sum_{(e_s,r,e_o,t)^{-}\in\mathcal{S}_{\tilde{e}}^{-}} \max\left(\gamma - s(e_s,r,e_o,t) + s(e_s,r,e_o,t)^{-},0\right), \tag{5}$$

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

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

$$\phi_{\tilde{e}} = \phi - \eta\frac{\partial\hat{\mathcal{L}}(f_\phi,\mathcal{S}_{\tilde{e}})}{\partial\phi}, \tag{6}$$

where η is the inner loop learning rate. By Eq. 6, we simulate the adaption for new entities. Updating from an optimal global parameter ϕ, 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 ϕ with a robust generalization ability over time.

Outer optimization: Since new entities evolve over time, it is not only $\mathcal{Q}_{\tilde{e}}$ but also each part of $\mathcal{Q}_{\tilde{e}}$ that follows different distributions. Thus, instead of optimizing the meta-learner on the whole query set $\mathcal{Q}_{\tilde{e}}$ at once by Eq. 2, we first adapt entity-specific parameters $\phi_{\tilde{e}}$ to recent facts, then gradually increase the difficulties by autoregressively adapting it to farther away future facts. Through this process, we are

Algorithm 2: MetaTKGR: Meta-training.

Input: Temporal knowledge graph $\mathcal{G}^T \subseteq \mathcal{E}^T\times\mathcal{R}\times\mathcal{E}^T\times\mathcal{T}$, randomly initialized ϕ.
Output: Learned global parameter ϕ.
Simulate new entity set $\tilde{\mathcal{E}}$ from $\mathcal{E}^T$;
Train model parameter ϕ from many-shot entities $\mathcal{E}^T\setminus\tilde{\mathcal{E}}$;
Construct $\mathcal{S}_{\tilde{e}}$ and $\mathcal{Q}_{\tilde{e}}$ for each new entity $\tilde{e}$, where $\mathcal{Q}_{\tilde{e}} = \{\mathcal{Q}_{\tilde{e}}^{(1)},\mathcal{Q}_{\tilde{e}}^{(2)},\cdots,\mathcal{Q}_{\tilde{e}}^{(M)}\}$;
while ϕ *not converge* **do**
  for *each time interval* m **do**
    # Outer optimization:
    for *each new entity* $\tilde{e}$ **do**
      # Inner optimization:
      Calculate $\hat{\mathcal{L}}(f_\phi,\mathcal{S}_{\tilde{e}})$ on $\mathcal{S}_{\tilde{e}}$ by Eq. 5;
      Perform adaptation by Eq. 6 to update $\phi_{\tilde{e}}$;
      Calculate $\hat{\mathcal{L}}(f_{\phi_{\tilde{e}}},\mathcal{Q}_{\tilde{e}}^{(m)})$ by Eq. 5;
    Calculate temporal adaptation regularizer by Eq. 28;
    Update $\phi^{(m)}$ by Eq. 9;
    $\phi\leftarrow\phi^{(m)}$

able to guide and stablize the training of global parameters via a temporal signal. Specifically, we split the time span of query sets into M equidistant time intervals, where each new entity has a sequence of query set corresponding to different time intervals $\mathcal{Q}_{\tilde{e}} = \{\mathcal{Q}_{\tilde{e}}^{(1)}, \mathcal{Q}_{\tilde{e}}^{(2)}, \cdots, \mathcal{Q}_{\tilde{e}}^{(M)}\}$. We first adapt and optimize $\phi_{\tilde{e}}$ for each new entity on $\mathcal{Q}_{\tilde{e}}^{(1)}$ for $\phi^{(1)}$ encoding global knowledge for predictions in the 1-st time interval. Then we gradually adapt each $\phi_{\tilde{e}}$ 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 [29], 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 predictive loss in new time interval with its empirical verson:

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

$$\mathcal{L}(f_{\phi^{(m)}}, \mathcal{D}^{(m+1)}) \leq \hat{\mathcal{L}}(f_{\phi^{(m)}}, \mathcal{D}^{(m+1)}) + \sqrt{\frac{\mathbb{KL}(q(\phi^{(m+1)})\|p(\phi^{(m)})) + \log\frac{|\mathcal{D}^{(m+1)}|}{\delta}}{2|\mathcal{D}^{(m+1)}|-1}}, \tag{7}$$

where $\mathcal{L}(f_{\phi^{(m)}}, \mathcal{D}^{(m+1)}) = \mathbb{E}_{\tilde{e}\sim p(\tilde{\mathcal{E}})}\left[\mathcal{L}(f_{\phi_{\tilde{e}}^{(m)}}, \mathcal{Q}_{\tilde{e}}^{(m+1)})\right]$ denotes real predictive loss on new time interval with new and unknown data distribution, $\hat{\mathcal{L}}(f_{\phi^{(m)}}, \mathcal{D}^{(m+1)})$ denote the empirical version of the loss, and $\mathbb{KL}(\cdot\|\cdot)$ 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*:

$$\mathcal{R}(f_{\phi^{(m)}}; \mathcal{D}^{(m+1)}) \triangleq \hat{\mathcal{L}}(f_{\phi^{(m)}}, \mathcal{D}^{(m+1)}) + \sqrt{\frac{\mathbb{KL}(q(\phi^{(m+1)})\|p(\phi^{(m)})) + \log\frac{|\mathcal{D}^{(m+1)}|}{\delta}}{2|\mathcal{D}^{(m+1)}|-1}}. \tag{8}$$

Remark. To interpret the temporal adaptation regularizer $\mathcal{R}(f_{\phi^{(m)}}; \mathcal{D}^{(m+1)})$, 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 ϕ 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 $m=M$:

$$\phi^{(m+1)} = \phi^{(m)} - \beta\frac{\partial\mathcal{R}(f_{\phi^{(m)}}; \mathcal{D}^{(m+1)})}{\partial\phi}, \tag{9}$$

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 $\tilde{e}$, we can first fine-tune ϕ on few-shot links, and then utilize $f_{\phi_{\tilde{e}}}$ to represent $\tilde{e}$ for future predictions.

4 Experiment

Datasets. We evaluate the proposed MetaTKGR framework on three public TKGs, where **YAGO** [28] 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.

Table 2: The results of 3-shot temporal knowledge graph reasoning. Average results on 5 independent runs are reported. $*$ indicates the statistically significant improvements over the best baseline, with p-value smaller than 0.001. We report standard deviation in Appendix A.5

Models	YAGO				WIKI				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	0.223	0.158	0.242	0.360	0.161	0.111	0.171	0.236	0.058	0.052	0.062	0.098
TransR	0.234	0.165	0.259	0.382	0.183	0.138	0.188	0.245	0.061	0.062	0.073	0.109
RotatE	0.241	0.182	0.278	0.409	0.232	0.171	0.223	0.284	0.078	0.074	0.082	0.128
RE-NET	0.261	0.210	0.298	0.410	0.261	0.210	0.251	0.331	0.232	0.139	0.241	0.369
LAN	0.230	0.154	0.247	0.352	0.185	0.133	0.201	0.287	0.207	0.119	0.234	0.321
I-GEN	0.303	0.238	0.323	0.420	0.221	0.179	0.229	0.264	0.212	0.120	0.251	0.346
T-GEN	0.292	0.218	0.310	0.394	0.234	0.185	0.222	0.271	0.169	0.122	0.184	0.265
MetaDyGNN	0.350	0.270	0.379	0.511	0.309	0.238	0.309	0.459	0.307	0.216	0.309	0.469
MetaTKGR	**0.370***	**0.303***	**0.416***	**0.558***	**0.329***	**0.253***	**0.335***	**0.489***	**0.335***	**0.249***	**0.340***	**0.527***
Gains (%)	*5.68*	*12.21*	*9.80*	*9.19*	*6.26*	*6.38*	*8.47*	*6.35*	*9.11*	*14.85*	*9.89*	*12.4*

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

Models	YAGO				WIKI				ICEWS18			
	1-shot		2-shot		1-shot		2-shot		1-shot		2-shot	
	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
TransE	0.183	0.268	0.193	0.304	0.144	0.186	0.146	0.213	0.049	0.077	0.058	0.086
TransR	0.189	0.270	0.198	0.312	0.160	0.183	0.160	0.225	0.050	0.080	0.060	0.090
RotatE	0.215	0.280	0.210	0.359	0.175	0.190	0.201	0.268	0.068	0.098	0.070	0.091
RE-NET	0.221	0.304	0.233	0.390	0.212	0.259	0.239	0.294	0.185	0.250	0.200	0.341
LAN	0.196	0.269	0.200	0.310	0.174	0.275	0.162	0.273	0.170	0.301	0.188	0.317
I-GEN	0.238	0.321	0.237	0.402	0.181	0.241	0.223	0.287	0.199	0.320	0.177	0.337
T-GEN	0.247	0.331	0.260	0.379	0.202	0.245	0.240	0.319	0.131	0.262	0.161	0.259
MetaDyGNN	0.269	0.396	0.316	0.496	0.241	0.371	0.271	0.390	0.249	0.420	0.269	0.441
MetaTKGR	**0.294***	**0.428***	**0.356***	**0.526***	**0.277***	**0.419***	**0.309***	**0.441***	**0.295***	**0.496***	**0.301***	**0.500***
Gains (%)	*9.43*	*8.04*	*12.69*	*6.14*	*14.64*	*12.93*	*14.04*	*13.20*	*18.45*	*17.87*	*11.47*	*13.39*

Table 1: Dataset statistics.

Datasets	# Entities	# Relations	# Quadruples	Time Unit
YAGO	10,623	10	201,090	1 year
WIKI	12,554	24	669,935	1 year
ICEWS18	23,033	256	281,205	1 day

Baselines. We compare eight state-of-the-art baselines from four related areas: 1) **TransE** [2], 2) **TransR** [27], 3) **RotatE** [41]: Translation distance based embedding methods for static knowledge graphs; 4) **RE-NET** [21]: A temporal knowledge graph embedding method; 5) **LAN** [46]; 6) **I-GEN** [1]; 7) **T-GEN** [1]: Few-shot methods on static knowledge graph; 8)**MetaDyGNN** [52]: 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 $(\tilde{e}, r, ?, t)$ or $(?, r, \tilde{e}, t)$, 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 [27, 41].

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 superiority 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 unsatisfied 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 3 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).

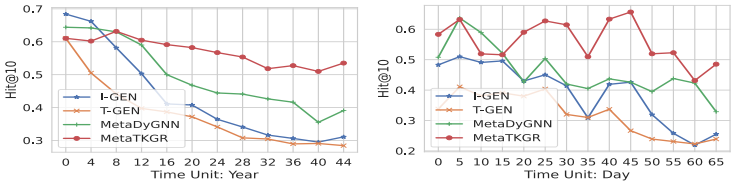

Figure 3: Performance of YAGO (left) and ICEWS18 (right) over time.

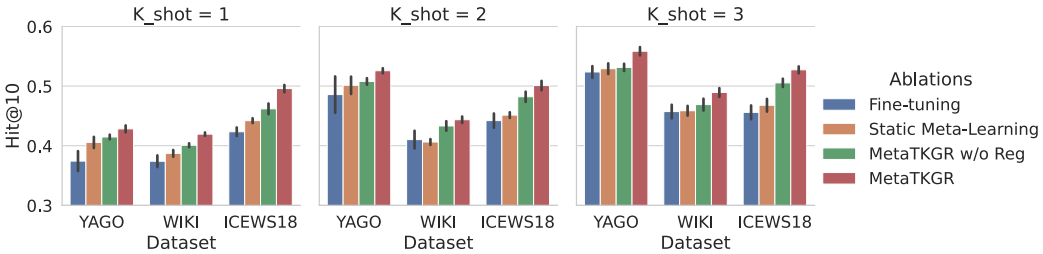

Figure 4: Ablation Studies, evaluated by filtered $Hit@10$.

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 conventional 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 4 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.

	1-Shot (Training)			3-Shot (Training)		
Test	**MRR**	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
1-S	0.294	0.240	0.428	0.281	0.242	0.403
3-S	0.354	0.297	0.540	0.370	0.303	0.558
5-S	0.377	0.360	0.569	0.389	0.369	0.591
R-S	0.358	0.304	0.542	0.377	0.316	0.570

Table 4: Cross-shot results of MetaTKGR on YAGO.

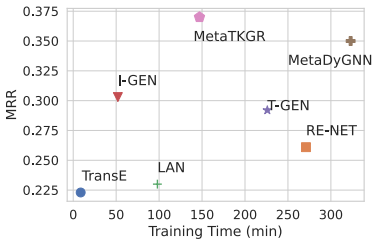

Figure 5: MRR over training time.

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.

Efficiency Analysis. To demonstrate the efficiency of our temporal meta-learning framework, we train MetaTKGR and baseline models from scratch on both existing entities and new entities and compare the training time, for the 3-shot experiment on YAGO. Figure 5 shows that MetaTKGR significantly outperforms baseline models with reasonable training time. Compared with slow temporal models *RE-NET, MetaDyGNN* for knowledge graph reasoning, MetaTKGR is more efficient because 1) our temporal encoder can learn temporal entity embeddings via sampled temporal neighbors at each continuous timestamp without using RNNs; 2) our temporal neighborhood sampler can prevent the size of multi-hop neighbors from increasing exponentially.

5 Related Work

Few-shot Learning on Knowledge Graph. Few-shot learning (FSL) [47] aims at learning generalized knowledge from existing tasks to extract transferable priors for new tasks with few labeled samples, including metric learning based approaches [45, 40] and meta-learning based approaches [10, 34, 11, 53]. For knowledge graph reasoning, the success of few-shot learning has facilitated learning the representations of scarce entities [16, 46, 38, 1] and scarce relations [55, 50, 6, 30] 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 [39]. 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 [17, 36]. [16] computes the representations of few-shot entities by GNN-based neighboring aggregation scheme. [46, 38, 30] 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 [1] explores how to mate-train the neighborhood aggregator so as to effectively adapt global knowledge for few-shot knowledge reasoning. More broadly on homogeneous graphs, Meta-Graph [4], G-Meta [20] and MetaDyGNN [52] are proposed to utilize meta-learning for link prediction across multiple graphs, on static/temporal homogeneous graphs respectively. However, most of the above 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 [52] is a very recent framework combining MAML with TGAT [8] 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 KGs with much 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]. Compared with static KG reasoning tasks, 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 [51, 14, 18], utilizing recurrent neural networks (RNNs) to adaptively learn the dynamic evolution from historical fact sequence [21, 48], or learning a sequence of evolving representations from discrete knowledge graph snapshots [19, 26, 24, 21]. 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 studied 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 proposed a novel Meta Temporal Knowledge Graph Reasoning framework, namely MetaTKGR. It extracts the global knowledge of sampling and aggregating temporal neighbors, which can be adapted 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 analyzed and proposed a temporal adaptation regularizer to stabilize and generalize the learned knowledge on future tasks. We empirically validated the effectiveness of MetaTKGR on three real-world temporal knowledge graphs, on which the proposed framework significantly outperforms an extensive set of SOTA baselines.

References

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

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2013.

[3] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data, 2015.

[4] Avishek Joey Bose, Ankit Jain, Piero Molino, and William L. Hamilton. Meta-graph: Few shot link prediction via meta learning. *CoRR*, 2019.

[5] Jie Chen, Tengfei Ma, and Cao Xiao. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018.

[6] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[7] Leyang Cui, Yu Wu, Shujie Liu, and Yue Zhang. Knowledge enhanced fine-tuning for better handling unseen entities in dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[8] da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. HyTE: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[11] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 2018.

[12] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[13] Yipeng Geng, Yali Shao, Shanwen Zhang, and Xiaoyun He. Multi-hop temporal knowledge graph reasoning over few-shot relations with novel method. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 551–556, 2022.

[14] Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. Diachronic Embedding for Temporal Knowledge Graph Completion. In *AAAI*, 2020.

[15] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 192–202, November 2016.

[16] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *IJCAI'17*, page 1802–1808, 2017.

[17] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, 2017.

[18] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*, 2021.

[19] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 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*, 2021.

[20] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *NeurIPS*, 2020.

[21] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *EMNLP*, 2020.

[22] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *WWW '18*, 2018.

[23] Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, November 2020.

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

[25] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. 2021.

[26] Siyuan Liao, Shangsong Liang, Zaiqiao Meng, and Qiang Zhang. Learning dynamic embeddings for temporal knowledge graphs. In *WSDM '21*, 2021.

[27] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187, 2015.

[28] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, 2015.

[29] David A. McAllester. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 164–170. ACM, 1999.

[30] Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[31] Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. EvoKG: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *WSDM*, 2022.

[32] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, Linlin Chen, Taeho Jung, and Junze Han. Social network de-anonymization and privacy inference with knowledge graph model. *IEEE Transactions on Dependable and Secure Computing*, pages 679–692, 2019.

[33] Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019.

[34] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[35] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, August 2021.

[36] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, pages 593–607, 2017.

[37] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[38] Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. Adaptive attentional network for few-shot knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1681–1691. Association for Computational Linguistics, 2020.

[39] Baoxu Shi and Tim Weninger. Open-world knowledge graph completion. *CoRR*, 2017.

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[41] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[42] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3462–3471, 2017.

[43] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.

[44] Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 2017.

[45] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[46] Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19, 2019.

[47] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 2020.

[48] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L. Hamilton. Temp: Temporal message passing for temporal knowledge graph completion, 2020.

[49] Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, September 2017.

[50] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[51] Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. Temporal knowledge graph completion based on time series gaussian embedding. In *The Semantic Web – ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I*, page 654–671, 2020.

[52] Cheng Yang, Chunchen Wang, Yuanfu Lu, Xumeng Gong, Chuan Shi, Wei Wang, and Xu Zhang. Few-shot link prediction in dynamic networks. In *WSDM '22*, 2022.

[53] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7045–7054, 2019.

[54] Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Andrey Malevich, Rajgopal Kannan, Viktor Prasanna, Long Jin, and Ren Chen. Decoupling the depth and scope of graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[55] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. Few-shot knowledge graph completion. In *AAAI*, 2019.

[56] 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.

[57] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1006–1014, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Our work studies few-shot temporal knowledge graph reasoning and all experimental datasets are anonymous. All algorithms, statements and conclusions in this paper have no negative social impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please see statement and discussion of Theorem 3.1 in Section 3 and Appendix A.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] We include complete proofs of all theoretical results in Appendix A.1 due to space limitation.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please see Section 4 and Appendix A.4 for reproducibility details. The code will be released upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 4 and Appendix A.4 for training details of our models and baseline models.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Average results on 5 independent runs are reported. Please see Section 4 and Appendix A.5 for error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Section 4 and Appendix A.5 for computational resources used in this paper. All experiments are run on the same server with 2 GPUs (GeForce RTX 3090) and CPUs (AMD Ryzen Threadripper 3970X 32-Core Processor).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please see Section 4 and Appendix A.2
   (b) Did you mention the license of the assets? [Yes] Please see Section 4 and Appendix A.5
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] All datasets used in experiments are publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Please see Section 4 and Appendix A.5, all experimental datasets are anonymous.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing or conducted research with human subjects.