# Question Calibration and Multi-Hop Modeling for Temporal Question Answering

**Chao Xue**[*1], **Di Liang**[*2], **Pengfei Wang**[3], **Jing Zhang**[1†]

[1]School of Software, Beihang University, Beijing, China
[2]School of Computer Science, Fudan University, Shanghai, China
[3]School of Software, Zhejiang University, Hangzhou, China
{xuechao, zhang_jing}@buaa.edu.cn, liangd17@fudan.edu.cn, wangpf@zju.edu.cn

## Abstract

Many models that leverage knowledge graphs (KGs) have recently demonstrated remarkable success in question answering (QA) tasks. In the real world, many facts contained in KGs are time-constrained thus temporal KGQA has received increasing attention. Despite the fruitful efforts of previous models in temporal KGQA, they still have several limitations. (I) They adopt pre-trained language models (PLMs) to obtain question representations, while PLMs tend to focus on entity information and ignore entity transfer caused by temporal constraints, and finally fail to learn specific temporal representations of entities. (II) They neither emphasize the graph structure between entities nor explicitly model the multi-hop relationship in the graph, which will make it difficult to solve complex multi-hop question answering. To alleviate this problem, we propose a novel **Q**uestion **C**alibration and **M**ulti-**H**op **M**odeling (**QC-MHM**) approach. Specifically, We first calibrate the question representation by fusing the question and the time-constrained concepts in KG. Then, we construct the GNN layer to complete multi-hop message passing. Finally, the question representation is combined with the embedding output by the GNN to generate the final prediction. Empirical results verify that the proposed model achieves better performance than the state-of-the-art models in the benchmark dataset. Notably, the Hits@1 and Hits@10 results of QC-MHM on the CronQuestions dataset's complex questions are absolutely improved by 5.1% and 1.2% compared to the best-performing baseline. Moreover, QC-MHM can generate interpretable and trustworthy predictions.


Figure 1: Examples of complex queries in both Google Search and ChatGPT yield incorrect results.

## Introduction

Knowledge graph question answering (KGQA) is a core technique in many NLP applications, such as search and recommendation (Huang et al. 2019; Xian et al. 2019; Guan et al. 2021). Among several branches of KGQA, temporal KGQA is a recently emerging direction and has shown great potential in real-world practices. There are critical differences between traditional KGQA and temporal KGQA tasks, which are summarized as follows: (I) Temporal KGQA has more complex semantic information, unlike the traditional KGs constructed based on the tuple of (subject, predicate, object)[1], temporal KGs are attached with additional timestamps. In other words, the tuple of temporal KGs is (subject, predicate, object, time duration). (II) Temporal KGQA is expected to generate answers with more diverse types. Different from regular KGQA whose answers are always entities, the answer of temporal KGQA can either be an entity or a timestamp. The above differences make the temporal KGQA more challenging.

To solve the above problem, the limited literature either decomposes the given question into non-temporal and temporal sub-question to answer (Jia et al. 2018; Kingma

[*]These authors contributed equally.
[†]Corresponding author.


[1]Some researchers refer to predicates as relations. The two are equivalent.

and Ba 2015) or directly combines the pre-trained language model with the temporal KG to generate answers (Saxena, Chakrabarti, and Talukdar 2021; Shuman et al. 2013). These methods can achieve satisfying performance on the questions with simple-entity or simple-time (refer to figure 1 for examples), but fail to answer the questions with complex templates. We argue that the current state-of-the-art methods have not been well solved, and may not even be aware of the following challenges which we address in this paper:

**Q1: How to capture implicit or explicit temporal information to calibrate question representation?** In previous methods, the question is usually encoded by PLMs. The result is that these methods will over-rely on the information of the entities involved in question, and ignore the entity shift caused by time constraints. Take the question *"Who was lasha talakhadze previous Olympic..."* depicted in Fig. 1 as an example. The Google search ignores the time constraint "previous" and purely regards "lasha talakhadze" and "Olympic" as the query, which leads to the wrong answers. However, there is rich temporal information in temporal KGs, which can promote understanding of the given question. Unfortunately, many prior kinds of research use KGs solely for querying the answer rather than enriching the question representations.

**Q2: How to effectively model multi-hop relationships between entities in temporal knowledge graph?** Existing methods almost don't emphasize the graph structure among entities in temporal KGs and fail to model multi-hop relational paths explicitly, which are not beneficial for reasoning, as demonstrated in previous research [Ren et al., 2020]. Such models struggle when answering the given questions requires multiple facts or multi-hop reasoning (i.e., the second example in Figure 1). Hence, integrating multi-hop information of KGs can facilitate complex question reasoning, which remains unexplored in temporal KGQA tasks. Moreover, these proposed methods lack certain transparency about their predictions, since they don't model the reasoning paths well and the whole process is invisible.

To address the aforementioned limitations, we propose an **Q**uestion **C**alibration and **M**ulti-**H**op **M**odeling (**QC-MHM**) approach for temporal KGQA. Our goal is to develop a reasoning model that can effectively infer answer entities for the given questions. Concretely, for a given question, we first *select* the relevant knowledge (*i.e.*, Subject-Predicate-Object (*abbr*. SPO)) of the entity in the question from TKGs. We design an *question calibration* mechanism to incorporate the SPO into the question representations, which allows the question embeddings to encode the relevant knowledge from temporal KGs, corresponding to **(Q1)**. Next, to build the timestamp embeddings with prior knowledge of the temporal order, we employ an auxiliary task for each pair of timestamp embeddings, which is crucial for further improvements in the model performance. Then, to address **(Q2)**, we explicitly leverage the structural information among entities of KGs via the graph neural networks (GNNs). Moreover, to directly model relational paths, we perform multi-hop message aggregation that allows each node to access its æ-hop neighbors within a single propagation layer, which is significantly superior to one-hop propagation. In modeling relational paths, we introduce an attention mechanism to score the reasoning path. In this way, our model can be interpreted according to this score while reasoning.

## Related Work

**Temporal KGQA**: Generally, KG embedding algorithms (Bordes et al. 2013; Yang et al. 2015; Trouillon et al. 2017) are employed to initialize entity and relation embeddings to help answer a question in the task of KGQA (Saxena, Tripathi, and Talukdar 2020). For temporal KGQA, we typically adopt temporal KG embedding approaches, such as TComplEx (Lacroix, Obozinski, and Usunier 2020), for initializing and also obtaining the timestamp embeddings. Recently, many researchers have focused on temporal KGQA and have proposed corresponding methods for this task (Jia et al. 2018; Saxena, Chakrabarti, and Talukdar 2021; Mavromatis et al. 2021; Jia et al. 2021; Shang et al. 2022a; Balcilar et al. 2020). Among these models, there are three representative ones: CronKGQA (Saxena, Chakrabarti, and Talukdar 2021), TMA (Liu et al. 2023; Shang et al. 2022b; Sharma et al. 2022) and TSQA (Shang et al. 2022b). CronKGQA utilizes recent advances in temporal KG embeddings and feeds the given questions to pre-trained LMs for answer prediction. Moreover, a dataset, CronQuestions, is proposed in this work, which is larger and more comprehensive than previous benchmarks that mainly employ hand-crafted templates to handle temporal information. Hence, we use CronQuestions as the evaluation dataset. TMA further extracts entity-related information from KGs and adopts a multi-way mechanism for information fusion. TSQA is equipped with a time estimation module that allows unwritten timestamps to be inferred from questions and presents a contrastive learning module that improves sensitivity to time relation words.

**Graph Neural Networks**: GNNs have attracted much attention due to their ability to model structured data and have been developed for various applications in practice (Kipf and Welling 2017; Veličković et al. 2018; Liang et al. 2019a; Song et al. 2022; Wang et al. 2022; Zhu et al. 2022; Yasunaga et al. 2022; Liang et al. 2019b; Zheng et al. 2022; Hasling, Clancey, and Rennels 1984; Garcia-Duran, Dumančić, and Niepert 2018). Among these models, graph convolutional network(Kipf and Welling 2017) is a pioneering work that designs a local spectral graph convolutional layer for learning node embeddings. GraphSAGE (Hamilton, Ying, and Leskovec 2017) generates node embeddings by learning an aggregator function that samples and aggregates features from the nodes' local neighborhoods. Graph Attention Network (Veličković et al. 2018) assigns different weights to different neighbors of a node to learn its representations by introducing self-attention mechanisms. Recently, several models (Feng et al. 2020; Zhang et al. 2018; Lukovnikov et al. 2017) have been designed to shift the power of GNNs to general QA tasks. However, these models merely use vanilla GNNs that adopt a one-hop neighbor aggregation mechanism, which may limit their expressiveness. Additionally, these models cannot be directly applied to our focused scenarios, *i.e.*, temporal KGQA.

## Problem Definition

**Temporal KGQA** aims to find suitable answers from KGs $\Omega = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ for given free-text questions. The answer is either an entity or a timestamp. Here, $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ represent the union sets of entities, relations, and timestamps, respectively. $\mathcal{F}$ denotes the set of facts in the form of quadruples $(s, r, o, t)$, where $s, r, o$ and $t$ are the subject, relation, object, and timestamp, respectively.

**Temporal KG embeddings** aim to learn low-dimensional embeddings based on the facts contained in the KG. Concretely, we embed $s, o \in \mathcal{E}, r \in \mathcal{R}$ and $t \in \mathcal{T}$ based on the predefined score function $\phi(\cdot)$ to obtain the corresponding embeddings $e_s, e_o, e_r, e_t \in \mathbb{R}^{2D}$. Typically, the valid fact $f = (s, r, o, t) \in \mathcal{F}$ is scored much higher than invalid facts $f' = (s', r', o', t') \notin \mathcal{F}$, *i.e.*, $\phi(s, r, o, t) \gg \phi(s', r', o', t')$.

## Proposed Methods

In this section, we present the details of the QC-MHM. which includes three key modules: *time-sensitive KG embedding*, *question calibration and multi-hop modeling*, and *answer prediction*. Next, we will elaborate on each module.

### Time-Sensitive KG Embedding

We start by obtaining the embeddings of the entity, relation, and timestamp in the temporal KG using a time-sensitive KG algorithm. TComplEx, a prevalent method, can produce high-quality temporal KG embeddings. Its score function is as follows:

$$\begin{aligned}\phi(e_s, e_r, \bar{e}_o, e_t) &= \mathbf{Re}(\langle e_s, e_r \odot e_t, \bar{e}_o \rangle) \\ &= \mathbf{Re}(\sum\nolimits_{d=1}^{2D} e_s[d] e_r[d] e_t[d] \bar{e}_o[d])\end{aligned} \tag{1}$$

where $\mathbf{Re}$ denotes the real part in the complex space and $\langle \cdot \rangle$ represents the multi-linear product operation. Additionally, $e_s, e_r, e_o, e_t$ are complex-valued embeddings and $\bar{e}_o$ is the complex conjugate of $e_o$. Due to the learning procedure of TComplEx, it is proficient at inferring missing facts in temporal KGs, such as $(s, r, ?, t)$ and $(s, r, o, ?)$, which is suitable for our scenarios. Therefore, in this work, we combine it with temporal order information to generate pre-trained temporal KG embeddings.

However, the vanilla TComplEx algorithm does not explicitly consider the sequential ordering information of timestamps, which is detrimental to reasoning based on temporal signals. For example, for the question "*Who was awarded the Ballon d'Or after Lionel Messi?*", the relevant facts are (*Lionel Messi, award received, Ballon d'Or, [2009, 2009]*) and (*Cristiano Ronaldo, award received, Ballon d'Or, [2013, 2013]*). In the embedding space, it is helpful to be aware that 2013 is later than 2009 when answering this question. Inspired by the usage of position embeddings (Vaswani et al. 2017; Jia et al. 2021; Yasunaga et al. 2021; Nt and Maehara 2019; Hu et al. 2020; Liu et al. 2022a,b; Ma et al. 2022), we inject temporal order information into timestamp embeddings via an auxiliary task while training temporal KGs. Specifically, we define the position embedding of the $k$-th timestamp $\mathbf{t}_k$ as follows:

$$\mathbf{t}_k(c) = \begin{cases} \sin(k/10000^{2i/2d}), \text{if } c = 2i \\ \cos(k/10000^{2i/2d}), \text{if } c = 2i+1 \end{cases} \tag{2}$$

where $2d$ is the dimension of timestamps and $c$ denotes the even or odd position in the $2d$-dimensional vector. We can obtain the position embedding $\mathbf{t}_k \in \mathbb{R}^{2d}$ via Eq. 2. This position encoding method has the properties of uniqueness (*i.e.*, different timestamps have different position embeddings) and sequential ordering (*i.e.*, it can reflect the relative positions among timestamps). Next, we adopt linear regression to obtain the probability of timestamp $m$ being ahead of timestamp $n$ for the given pair $(m, n)$. A binary cross-entropy objective function is employed in this auxiliary task. The concrete formulas are as follows:

$$\begin{aligned}\rho(m, n) &= \sigma(\mathbf{W}_{ts}^{\top}((e_m + \mathbf{t}_m) - (e_n + \mathbf{t}_n))) \\ \mathcal{L}_{ts}(m, n) &= -\alpha(m, n) \log(\rho(m, n)) \\ &\quad - (1 - \alpha(m, n)) \log(1 - \rho(m, n))\end{aligned} \tag{3}$$

where $\sigma(\cdot)$ and $\mathbf{W}_{ts}$ are the sigmoid function and learnable parameters. $e_*$ and $\mathbf{t}_*$ are timestamp embeddings and the corresponding position embeddings. $\alpha(m, n)$=1 if $m < n$ and 0 otherwise, and $\rho(m, n)$ is the predicted probability of the time order. The final loss function of this module is the weighted sum of the loss function of TComplEx and the auxiliary task, *i.e.*, $\mathcal{L}_{fin} = \mathcal{L}_{tc} + \lambda \mathcal{L}_{ts}$, where $\lambda$ is the weight coefficient and $\mathcal{L}_{tc}$ is the margin loss function as referred to in TComplEx.

### Question Calibration and Multi-Hop Modeling

This module aims to calibrate the question representation by combining semantic information from PLMs and temporal knowledge graphs, while utilizing GNN for multi-hop relation perception.

**(I) Question Calibration**: In order to overcome the entity transfer caused by time constraints, first, we use Sentence-BERT (Reimers and Gurevych 2019) to find relevant SPO as recall candidates for potential entity transfer to the question. The tokenized question is fed to the sentenceBert to obtain token embeddings. SPO information of temporal KG is also performed in the same operations as above, and we can get the SPO embeddings $S$. The concrete formulas are as Eq. 4.

$$\begin{aligned} Q &= \text{sentenceBert}(question) \\ S &= \text{sentenceBert}(< \text{SPO} >) \end{aligned} \tag{4}$$

In general, we take the [CLS] embedding (*i.e.*, $q_{[\text{CLS}]}$ and $q_{s_{[\text{CLS}]}}$) as the final question embedding and SPO embedding.And we apply the cosine similarity on the question and SPO representation to learn the matching scores as follows:

$$score(q_{[\text{CLS}]}, q_{s_{[\text{CLS}]}}) = \frac{q_{[\text{CLS}]}^{\top} q_{s_{[\text{CLS}]}}}{\|q_{[\text{CLS}]}\| \|q_{s_{[\text{CLS}]}}\|} \tag{5}$$

where $score$ is a scalar. The top ten scored SPOs are selected as candidate information. Then, previous studies (Rocktäschel et al. 2015; Tan et al. 2018; Xue et al. 2023)

demonstrate the effectiveness of word-level attention in sentence pair modeling. Inspired by this, in order to model the relationship between questions and SPO from different perspectives, we design a multi-view alignment module, which uses different types of attention functions to compare the correlation of questions and SPO from different perspectives.

For a given question, we embed it with Eq. 4, excluding the [CLS] token, *i.e.*, $\bar{Q} = [q_1, q_2, \ldots, q_n]$. For the ten selected SPOs, we take the [CLS] token of each SPO and concatenate them together, *i.e.*, $P = [S_1, S_2, \ldots, S_m]$ ($m$ is the number of selected SPOs). Then, the candidate SPOs can be matched by the word at each position $k$ of the question. which are formulated as follows:

$$\tilde{p}_k^{\ell} = \Phi_{\ell}(P, q_k; \mathbf{W}_{\ell}) \tag{6}$$

where $\tilde{p}_k^{\ell}$ is the corresponding weighted-sum representation of SPOs specified by $q_k$, employing the attention function $\Phi_{\ell}$ with parameterized by $\mathbf{W}_{\ell}$, in which $\ell$ denotes concat attention, dot attention, and minus attention, respectively. More precisely, the different attention mechanisms can be described as follows:

**Concat Attention**:

$$\begin{aligned} h_j^k &= v_{cat}^{\top} \tanh(\mathbf{W}_{cat}[q_k, S_j]) \\ \alpha_i^k &= \exp(h_i^k) / \sum\nolimits_{j=1}^{m} \exp(h_j^k), \quad \tilde{p}_k^{cat} = \sum\nolimits_{i=1}^{m} \alpha_i^k S_i \end{aligned} \tag{7}$$

**Dot Attention**:

$$\begin{aligned} h_j^k &= v_{dot}^{\top} \tanh(\mathbf{W}_{dot}(q_k \odot S_j)) \\ \alpha_i^k &= \exp(h_i^k) / \sum\nolimits_{j=1}^{m} \exp(h_j^k), \quad \tilde{p}_k^{dot} = \sum\nolimits_{i=1}^{m} \alpha_i^k S_i \end{aligned} \tag{8}$$

**Minus Attention**:

$$\begin{aligned} h_j^k &= v_{min}^{\top} \tanh(\mathbf{W}_{min}(q_k - S_j)) \\ \alpha_i^k &= \exp(h_i^k) / \sum\nolimits_{j=1}^{m} \exp(h_j^k), \quad \tilde{p}_k^{min} = \sum\nolimits_{i=1}^{m} \alpha_i^k S_i \end{aligned} \tag{9}$$

Next, to obtain the attention-based question representation $\tilde{Q}^{\ell}$, we aggregate the matching information $\tilde{p}_k^{\ell}$ together with the word representation $q_k$ via the concatenation operation, *i.e.*, $\tilde{q}_k^{\ell} = [q_k, \tilde{p}_k^{\ell}]$. Finally, the linear transformation is applied to $\tilde{Q}^{\ell}$ that fuses the SPO information, *i.e.*, $\mathcal{Q}_{final} = \mathbf{W}[\tilde{Q}^{cat}, \tilde{Q}^{dot}, \tilde{Q}^{min}] = [\hat{q}_1, \ldots, \hat{q}_n]$. Similarly, the question can be matched by a particular SPO by performing the above multiway operation and linear transformation. In this way, we can obtain the updated SPO representation $\hat{S}_i$.

Finally, we use adaptive fusion to make question representations more time-aware, we use a gate mechanism to adaptively fuse the temporal information from SPOs.

$$\begin{aligned} \tilde{S} &= \tanh(\mathbf{W}_{\hat{S}_i} \frac{1}{m} \sum\nolimits_{i=1}^{m} \hat{S}_i + b_{\hat{S}_i}) \\ g_i &= \sigma(\mathbf{W}_{g_i}(\hat{q}_i \cdot \tilde{S})), \quad q_{\text{sem}} = g_i \hat{q}_i + (1 - g_i)\tilde{S} \end{aligned} \tag{10}$$

where $\sigma$ denote the nonlinear activation function, respectively. $q_{\text{sem}}$ is the final embedding of the word in the question, which is the representation containing the potential entity transfer information.

**(II) Multi-Hop Modeling**: We first construct a graph $G = (V, E)$ based on given temporal KGs, where $V$ is the set of nodes, denoting entities, and $E$ is the set of edges, connecting the triplet's subject and object. The value of an edge is the concatenation of a relation and timestamp, *i.e.*, $r||t$. The idea is to propagate both relations and timestamps via graph structures, which is specific to temporal KGQA tasks. The node and edge features can be initialized by the pre-trained time-sensitive KG encoder.

Next, we obtain annotated entities $\{\text{ent}_1, \text{ent}_2, \cdots, \text{ent}_w\}$ from each question $q$. For each entity $\text{ent}_i$, we then extract its æ-hop sub-graph $G_i$. The final relevant æ-hop sub-graph $G_q$ for the question can be obtained by combining each entity's sub-graph, *i.e.*, $G_q = \cup_{i=1}^{w} G_i$. Note that we restrict the answer selection to $G_q$ via the latent sub-graph extraction procedure, which can greatly reduce the search space and effectively facilitate the training process.

To directly leverage the structural information among entities of temporal KGs, we apply GNNs to the extracted subgraph. Typically, the classic message passing paradigm of GNNs can be formulated as:

$$\begin{aligned} a_v^{\ell} &= \textbf{AGGREGATE}(\{h_u^{\ell-1} : u \in \mathcal{N}_v\}) \\ h_v^{\ell} &= \textbf{COMBINE}(h_v^{\ell-1}, a_v^{\ell}) \end{aligned} \tag{11}$$

where $\mathcal{N}_v$ is the set of node $v$'s neighbors. $a_v^{\ell}$ is the aggregated message at layer $\ell$, and $h_v^{\ell}$ is node $v$'s embeddings at layer $\ell$ obtained by combining $h_v^{\ell-1}$ and $a_v^{\ell}$. However, in the above framework, the nodes in the graph can only access their one-hop neighbors through a single graph layer. In other words, suppose two nodes are not directly connected, they can only interact with each other by stacking a sufficient number of layers, which severely limits the capability of GNNs to explore the relationships between disjoint nodes.

To address this problem, we adopt a multi-hop message passing mechanism that works on all possible paths between two nodes. The first step is to compute the normalized attention using Eq. 12.

$$\begin{aligned} \mathscr{A}_{irj}^{\ell} &= \begin{cases} \beta^{\ell} \delta(\mathbf{W}_{ad}^{\ell}(h_i^{\ell}||h_j^{\ell}||h_r||h_t)), (v_i, r, v_j, t) \in G \\ -\infty, \text{otherwise} \end{cases} \\ \mathbf{A}^{\ell} &= \text{softmax}(\mathscr{A}^{\ell}) \end{aligned} \tag{12}$$

where $\beta^{\ell}$ and $\mathbf{W}_{ad}^{\ell}$ are the learnable weights shared by the $\ell$-th layer. $h_i^{\ell}$ is the embedding of node $i$, initialized by $h_i^0 = e_i$. $h_r$ and $h_t$ are the embeddings of relation $r$ and timestamp $t$, respectively. $\delta$ denotes the ReLU activation function. $\mathscr{A}^{\ell}$ and $\mathbf{A}^{\ell}$ represent the attention matrix obtained by applying the edges appearing in $G$ and the normalized attention matrix derived by performing a row-wise softmax function, respectively. In addition, since paths with different importance are assigned corresponding weights using Eq. 12, we can derive the reasoning path based on these weights. $||$ denotes the concatenation operation. To enable the aggregation of multi-hop messages to a target node within a single propagation layer, we employ a mechanism defined as follows:

$$\mathbf{D} = \sum\nolimits_{\tau=0}^{\aleph} \xi_{\tau} \mathbf{A}^{(\tau)} \tag{13}$$

where $\xi_\tau$ are trainable vectors. $\mathbf{A}^{(\tau)}$ is the powers of $\mathbf{A}$, which considers relational paths with length limits up to $\tau$ from neighboring nodes to the target node. In other words, the target node's context (*i.e.*, intermediate neighbors) and its local graph structure are involved in attention calculation. This procedure successfully creates attentional interactions between a node and its disjoint neighbors beyond one-hop. In practice, we can achieve impressive performance when empirically setting the diffusion distance $\aleph \in [2, 4]$ since many graphs have small-world properties with lower diameters. Subsequently, the transition matrix $\mathbf{D}$ is leveraged to update the nodes' embeddings to obtain $\mathbf{H}^{\ell+1}$ in Eq. 14.

$$\mathbf{H}^{\ell+1} = \mathbf{D}\mathbf{H}^{\ell} \tag{14}$$

Finally, we perform an average pooling operation on the nodes of the extracted sub-graph to acquire the question's multi-hop information $\mathbf{Q}_{mlh}$, formulated as Eq. 15.

$$\mathbf{Q}_{mlh} = \frac{1}{|V_q|} \sum\nolimits_{i \in V_q} h_i^L \tag{15}$$

where $V_q$ is the node set of the sub-graph and $h_i^L$ is the node embeddings at the $L$-th layer.

To better integrate the question's local and global information, we employ a sophisticated knowledge fusion layer $\Phi(\cdot)$, that contains several Transformer encoder layers. After performing the Transformer-based information fusion layer, we obtain the final question representation, *i.e.*, $\mathbf{Q}_{fin} = \Phi(\mathbf{Q}_{sem}||\mathbf{Q}_{mlh})$.

## Answer Prediction

We use two-layer MLPs to transform $\mathbf{Q}_{fin}$ into $\mathbf{Q}_{ent}$ and $\mathbf{Q}_{tim}$, which correspond to entity and timestamp prediction, respectively and are defined in Eq. 16.

$$\begin{aligned} \mathbf{Q}_{ent} &= \mathbf{MLP}(\mathbf{Q}_{fin}) \\ \mathbf{Q}_{tim} &= \mathbf{MLP}(\mathbf{Q}_{fin}) \end{aligned} \tag{16}$$

Next, we define an entity score function $\phi_{ent}(\cdot)$ and a timestamp score function $\phi_{tim}(\cdot)$ to obtain the scores of candidate entities and timestamps, as shown in Eq. 17.

$$\begin{aligned} \phi_{ent}(\tilde{\varepsilon}) &= \mathbf{Re}(\langle e_s, \mathbf{Q}_{ent} \odot e_t, \bar{e}_{\tilde{\varepsilon}} \rangle) \\ \phi_{tim}(\tilde{t}) &= \mathbf{Re}(\langle e_s, \mathbf{Q}_{tim} \odot e_{\tilde{t}}, \bar{e}_o \rangle) \end{aligned} \tag{17}$$

where $\tilde{\varepsilon} \in \mathcal{E}_q$ and $\tilde{t} \in \mathcal{T}_q$, in which $\mathcal{E}_q \subseteq \mathcal{E}$ and $\mathcal{T}_q \subseteq \mathcal{T}$ are specified by the sub-graph $G_q$ with respect to the given question $q$.

Finally, we concatenate the obtained scores for the entities and timestamps and perform the softmax function over them to obtain the answer probability. The objective function is the cross-entropy loss, as shown in Eq. 18.

$$\begin{aligned} \hat{y}_i &= \text{softmax}(\phi_{ent}(\cdot)||\phi_{tim}(\cdot)) \\ \mathcal{L}_{predict} &= -\sum\nolimits_i y_i \log(\hat{y}_i) \end{aligned} \tag{18}$$

where $y_i$ is the true answer to the question.

# Datasets and Baselines

## Datasets

We employ two temporal KGQA benchmarks, CRONQUESTIONS (Saxena, Chakrabarti, and Talukdar 2021) and Time-Questions (Jia et al. 2021). **CRONQUESTIONS** is the largest known dataset and is widely used in previous studies. It contains two main parts: the Wikidata temporal KG and a collection of free-text questions. Among them, there are 125K entities, 1.7K timestamps, 203 relations, and 328K facts in the form of quadruples in the temporal KG. Additionally, 410K unique question-answer pairs are given, where each question contains annotated entities and timestamps. Moreover, this dataset can be divided into entity and time questions based on the type of answers. It can also be divided into simple reasoning (*i.e.*, Simple Entity and Simple Time) and complex reasoning (*i.e.*, Before/After, First/Last, and Time Join) based on the questions' difficulty. **TimeQuestions** is another challenging dataset, which has 16k manually tagged temporal questions and are divided into four categories (*i.e.,* Explicit, Implicit, Temporal, and Ordinal) according to the type of time reasoning.

## Baselines

**Pre-Trained LMs** 1) BERT(Devlin et al. 2018): It employs the bidirectional Transformer to encode a large-scale corpus. In our experiments, we use BERT to produce the question embeddings and concatenate them with pre-trained entity and timestamp embeddings to predict the answer. 2) RoBERTa(Liu et al. 2019): It extends BERT by using a dynamic mask. Moreover, it is trained with a larger batch size and corpus. To answer each given question, we utilize RoBERTa to generate question embeddings and concatenate them with pre-trained entity and timestamp embeddings. 3) KnowBERT (Peters et al. 2019): It is a variant of BERT that introduces information from KGs, such as WikiData and WordNet. The [CLS] token is adopted in answer prediction. 4) T5 (Raffel et al. 2020): It proposes a powerful architecture called Text-to-text. To apply T5 to temporal questions, we transform each question into the form "temporal question: question?".

**General KG Embedding-Based Models** 1) EaE (Févry et al. 2020): It is an entity-aware method integrating entity knowledge into Transformer-based LMs. 2) EmbedKGQA (Saxena, Tripathi, and Talukdar 2020): It leverages the ComplEx method to produce KG embeddings for KGQA tasks and can only deal with non-temporal questions. To accommodate this task, random timestamp embeddings are used during the training stage.

**Temporal KG Embedding-Based Models** 1) CronKGQA (Saxena, Chakrabarti, and Talukdar 2021): It further extends EmbedKGQA by employing temporal KG embeddings and is designed for TKGQA tasks. 2) TMA (Fang and Liu 2022; Fei et al. 2022; Gui et al. 2018): It extracts the relevant SPO information and adopts multi-way attention to enhance question understanding. 3) TSQA (Shang et al. 2022b): It proposes a time-sensitive module to infer the time and contrastive losses to improve the model's ability to perceive time relation words, achieving promising performance.

| Model | Hits@1 | | | | | Hits@10 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Overall | Question Type | | Answer Type | | Overall | Question Type | | Answer Type | |
| | | Complex | Simple | Entity | Time | | Complex | Simple | Entity | Time |
| BERT | 0.243 | 0.239 | 0.249 | 0.277 | 0.179 | 0.620 | 0.598 | 0.649 | 0.628 | 0.604 |
| RoBERTa | 0.225 | 0.217 | 0.237 | 0.251 | 0.177 | 0.585 | 0.542 | 0.644 | 0.583 | 0.591 |
| KnowBERT | 0.226 | 0.220 | 0.238 | 0.252 | 0.177 | 0.586 | 0.539 | 0.646 | 0.582 | 0.592 |
| T5-3B | 0.252 | 0.240 | 0.251 | 0.283 | 0.180 | - | - | - | - | - |
| EmbedKGQA | 0.288 | 0.286 | 0.290 | 0.411 | 0.057 | 0.672 | 0.632 | 0.725 | 0.85 | 0.341 |
| T-EaE-add | 0.278 | 0.257 | 0.306 | 0.313 | 0.213 | 0.663 | 0.614 | 0.729 | 0.662 | 0.665 |
| T-EaE-replace | 0.288 | 0.257 | 0.329 | 0.318 | 0.231 | 0.678 | 0.623 | 0.753 | 0.668 | 0.698 |
| CronKGQA | 0.647 | 0.392 | 0.987 | 0.699 | 0.549 | 0.884 | 0.802 | 0.992 | 0.898 | 0.857 |
| TMA | 0.784 | 0.632 | 0.987 | 0.792 | 0.743 | 0.943 | 0.904 | 0.995 | 0.947 | 0.936 |
| TSQA | 0.831 | 0.713 | 0.987 | 0.829 | 0.836 | 0.980 | 0.968 | 0.997 | 0.981 | 0.978 |
| TempoQR | 0.918 | 0.864 | 0.990 | 0.926 | 0.903 | 0.978 | 0.978 | 0.993 | 0.980 | 0.974 |
| CTRN | 0.920 | 0.869 | 0.900 | 0.921 | 0.917 | 0.980 | 0.970 | 0.993 | 0.982 | 0.976 |
| QC-MHM (ours) | **0.971** | **0.946** | **0.992** | **0.962** | **0.966** | **0.992** | **0.986** | **0.998** | **0.993** | **0.987** |
| Ab. Imp | 5.1% | 7.7% | 0.2% | 3.6% | 4.9% | 1.2% | 1.6% | 0.1% | 1.1% | 0.9% |

Table 1: Performance of baselines and our methods on the CronQuestions dataset.

| Model | Overall | Explicit | Implicit | Temporal | Ordinal |
|---|---|---|---|---|---|
| CronKGQA | 0.393 | 0.388 | 0.380 | 0.436 | 0.332 |
| TMA | 0.436 | 0.442 | 0.419 | 0.476 | 0.352 |
| TempoQR | 0.459 | 0.503 | 0.442 | 0.458 | 0.367 |
| CTRN | 0.465 | 0.469 | 0.446 | 0.512 | 0.382 |
| QC-MHM | 0.531 | 0.533 | 0.508 | 0.607 | 0.401 |

Table 2: Hits@1 for different models on TimeQuestions.

# Results

## Model Performance

We present the results of our proposed QC-MHM and baselines on CRONQUESTIONS in terms of Hits@1 and Hits@10 in Table 1. QC-MHM achieves the best performance in all experimental settings, indicating its superiority on the temporal KGQA task. Remarkably, QC-MHM significantly outperforms the second-best model on both datasets. It achieves 7.7% and 4.9% absolute improvements on Hits@1 with respect to complex reasoning and time questions on CRONQUESTION, respectively. It also performs far better than other models for various types of questions in the TimeQuestions dataset in Table 2. For example, it achieves absolute improvements of 3.0% and 6.2% on Hits@1 for questions involving 'Explicit' and 'Implicit' types. While in the 'Temporal' type of questions, an absolute improvement of 9% is obtained over the best baseline.

We find that PLMs (*e.g.*, BERT) achieve unsatisfactory performance in this scenario, lagging far behind the TKG embedding-based models. A plausible reason is that these models do not introduce KG into this task, which is detrimental to question understanding. Despite the relative success of general KG embedding-based models (*e.g.*, EmbedKGQA) in common QA tasks, they still perform worse than TKG embedding-based models(*e.g.*,TMA, and TSQA) in our focused scenario. A possible reason is that they do not explicitly leverage temporal KG and neglect temporal information, which is crucial for the temporal KGQA task.

| Category | Complex Question | | | Simple Question | | All |
|---|---|---|---|---|---|---|
| | Before/ After | First/ Last | Time Join | Simple Entity | Simple Time | |
| EmbedKGQA | 0.199 | 0.324 | 0.223 | 0.421 | 0.087 | 0.288 |
| T-EaE-add | 0.256 | 0.285 | 0.175 | 0.296 | 0.321 | 0.278 |
| T-EaE-replace | 0.256 | 0.288 | 0.168 | 0.318 | 0.346 | 0.288 |
| CronKGQA | 0.288 | 0.371 | 0.511 | 0.988 | 0.985 | 0.647 |
| TMA | 0.581 | 0.627 | 0.675 | 0.988 | 0.987 | 0.784 |
| TSQA | 0.504 | 0.721 | 0.799 | 0.988 | 0.987 | 0.831 |
| TempoQR | 0.714 | 0.853 | 0.978 | 0.988 | 0.987 | 0.918 |
| CTRN | 0.747 | 0.880 | 0.897 | 0.991 | 0.987 | 0.920 |
| QC-MHM | **0.905** | **0.938** | **0.992** | **0.989** | **0.996** | **0.971** |

Table 3: Hits@1 for different question types.

| Model | Hits@1 | | | | |
|---|---|---|---|---|---|
| | Overall | Question Type | | Answer Type | |
| | | Complex | Simple | Entity | Time |
| QC-MHM | 0.971 | 0.946 | 0.992 | 0.962 | 0.966 |
| w/o Time Order | 0.902 | 0.884 | 0.949 | 0.883 | 0.916 |
| w/o Multi-Hop | 0.868 | 0.841 | 0.903 | 0.854 | 0.909 |
| w/o Calibration | 0.826 | 0.753 | 0.872 | 0.738 | 0.871 |

Table 4: Results of ablation studies on our model.

We present the Hits@1 results of our model and other baselines on different types in Table 3. QC-MHM is significantly superior to other models, especially for complex questions. Our model gains 15.8%, 5.8%, and 1.4% absolute improvement over "Before/After", "First/Last" and "Time Join", respectively, due to the consideration of the timestamp order and multi-hop structural information of the TKG.

## Ablation Study

To evaluate the contribution of each module in our framework, we perform extensive ablation experiments. The experimental results are shown in Table 4. (I) *W/o Time Order*: We exclude temporal order encoding and use the vanilla TComplEx method. (II) *W/o Multi-Hop*: We use the one-hop

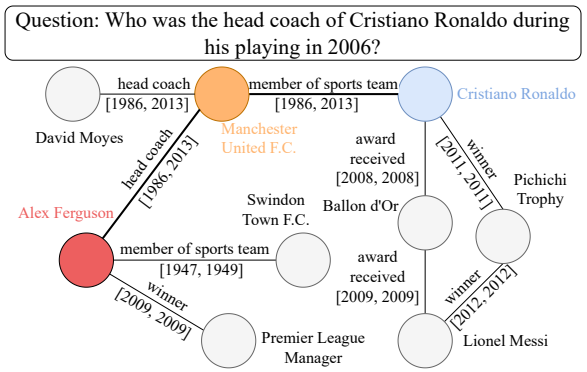


Figure 2: Visualization of a case study of the interpretability of our model. For brevity, we only show the key entities.

attention computed from the neighbors without multi-hop attention, similar to GAT. (III) *W/o Question Calibration*: We removed the module of Question Calibration.

The results are presented in Table 4. We can obtain the following insights: First, after eliminating the Question Calibration, the model's performance drops drastically, which is in line with our expectations. This result indicates that this module can provide helpful contextual information for accurately understanding the question. Second, the performance declines when we perform one-hop message passing instead of multi-hop, empirically demonstrating that multi-hop message passing is more expressive. Finally, complex questions require the temporal order information to be captured, thus removing this information inevitably harms the model.

## Interpretability of Multi-Hop Modeling

To interpret our model's reasoning process, we investigate the relational path attention weights induced by the attention layer of GNNs described in Eq. 12. Specifically, we trace high attention weights from entity nodes to the candidate answer nodes on the retrieved sub-graph $G_q$ by leveraging Best First Search (BFS). Fig. 2 illustrates one example. In this example, we note that the reasoning path contains "Cristiano Ronaldo" in the question and "Alex Ferguson" and "Manchester United F.C." in KGs. QC-MHM can make accurate predictions, *i.e.*, "Alex Ferguson", given the question. Notably, QC-MHM promotes rational reasoning by introducing

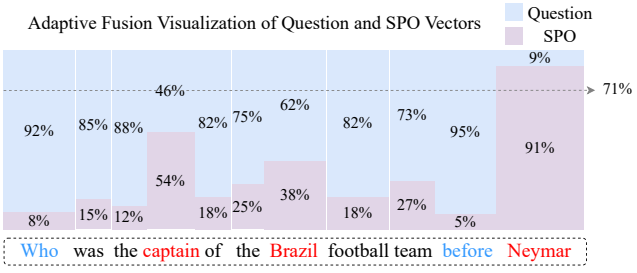


Figure 3: Visualization of the question and SPO vectors using the adaptive fusion technique.

| | |
|---|---|
| 0.00 | <Neymar, awarded, South American footballer, 2012> |
| 0.02 | <Brazil national team, awarded, World Cup champion, 1970> |
| 0.00 | <Miranda, joined, Brazil national team, 2007> |
| 0.01 | <Neymar, awarded, FIFA Ference Puskás Award, 2012> |
| 0.84 | <Neymar, position held, captain of Brazil national team, 2014> |
| 0.00 | <Dunga, coach, Brazil national team, 2014> |
| 0.00 | <Neymar, joined, Barcelona, 2013> |
| 0.11 | <Brazil national team, awarded, World Cup champion, 1994> |
| 0.01 | <Brazil national team, awarded, Confederations Cup, 2005> |
| 0.00 | <Pimenta, coach, Brazil national team, 1937> |
| 0.01 | <Neymar, joined, Paris Saint-Germain, 2017> |

Figure 4: Visualization of the SPO gradients. The left if percentage of gradients, and the right is details of the SPOs.

"Manchester United F.C.", which is not mentioned in the question, revealing the importance of background knowledge. It provides an interpretable reasonable path "Cristiano Ronaldo→Manchester United F.C.→Alex Ferguson".

## Interpretability of Question Calibration

To validate the role of SPO information selection, we use the question "Who was the captain of the Brazil national team before Neymar?" for a quantitative study. First, we take the hidden states of each SPO separately and then calculate their gradients. Finally, a normalization function is applied to obtain each SPO proportion. As shown in Fig. 4, the more helpful the SPO is in understanding the question, the higher its gradient proportion is., indicating that QC-MHM can capture useful inference information from SPOs. And, To verify the ratio of SPO and question fusion, Fig. 3 visualizes each word weight in the question and its information fusion ratio. QC-MHM decreases the information share of the entity word "Neymar" and increases the information from SPO. The information distribution is consistent with human cognition, indicating that QC-MHM mitigates the influence of the entities in the question while effectively fusing the SPO information with the question information adaptively.

## Conclusion

In this work, we propose the Question Calibration and Multi-Hop Modeling approach for the temporal KGQA task. Three specific modules are introduced to significantly improve the performance of the model. Specifically, a time-sensitive KG embedding module is used to add temporal ordering information. In addition, the question calibration and multi-hop modeling module adaptively fuse the SPO in the graph and explicitly model the multi-hop question, and finally gets the answer in the answer prediction module. Extensive experiments on two widely used datasets show that QC-MHM achieves consistent improvements over baselines.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2021YFB1714300) and National Natural Science Foundation of China (No.62006012, No.62132001)

## References

Balcilar, M.; Renton, G.; Héroux, P.; Gaüzère, B.; Adam, S.; and Honeine, P. 2020. Analyzing the expressive power of graph neural networks in a spectral perspective. In *ICLR*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fang, F.; and Liu, Y. 2022. Time-aware Multiway Adaptive Fusion Network for Temporal KGQA. In *ISWC*.

Fei, Z.; Zhang, Q.; Gui, T.; Liang, D.; Wang, S.; Wu, W.; and Huang, X.-J. 2022. CQG: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6896–6906.

Feng, Y.; Chen, X.; Lin, B. Y.; Wang, P.; Yan, J.; and Ren, X. 2020. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In *EMNLP*.

Févry, T.; Soares, L. B.; Fitzgerald, N.; Choi, E.; and Kwiatkowski, T. 2020. Entities as Experts: Sparse Memory Access with Entity Supervision. In *EMNLP*.

Garcia-Duran, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 4816–4821.

Guan, R.; Liu, Y.; Feng, X.; and Li, X. 2021. Vpalg: Paper-publication prediction with graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 617–626.

Gui, T.; Zhang, Q.; Gong, J.; Peng, M.; Liang, D.; Ding, K.; and Huang, X.-J. 2018. Transferring from formal newswire domain with hypernet for twitter pos tagging. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2540–2549.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

Hasling, D. W.; Clancey, W. J.; and Rennels, G. 1984. Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies*, 20(1): 3–19.

Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; and Sun, Y. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1857–1867.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *WSDM*.

Jia, Z.; Abujabal, A.; Saha Roy, R.; Strötgen, J.; and Weikum, G. 2018. Tequila: Temporal question answering over knowledge bases. In *CIKM*.

Jia, Z.; Pramanik, S.; Saha Roy, R.; and Weikum, G. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 792–802.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Lacroix, T.; Obozinski, G.; and Usunier, N. 2020. Tensor decompositions for temporal knowledge base completion. In *ICLR*.

Liang, D.; Zhang, F.; Zhang, Q.; and Huang, X.-J. 2019a. Asynchronous deep interaction network for natural language inference. 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)*, 2692–2700.

Liang, D.; Zhang, F.; Zhang, W.; Zhang, Q.; Fu, J.; Peng, M.; Gui, T.; and Huang, X. 2019b. Adaptive multi-attention network incorporating answer information for duplicate question detection. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 95–104.

Liu, L.; Du, B.; Xu, J.; Xia, Y.; and Tong, H. 2022a. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1098–1108.

Liu, Y.; Li, M.; Li, X.; Giunchiglia, F.; Feng, X.; and Guan, R. 2022b. Few-shot Node Classification on Attributed Networks with Graph Meta-learning. In *SIGIR*.

Liu, Y.; Liang, D.; Fang, F.; Wang, S.; Wu, W.; and Jiang, R. 2023. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lukovnikov, D.; Fischer, A.; Lehmann, J.; and Auer, S. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, 1211–1220.

Ma, R.; Tan, Y.; Zhou, X.; Chen, X.; Liang, D.; Wang, S.; Wu, W.; Gui, T.; and Zhang, Q. 2022. Searching for optimal subword tokenization in cross-domain ner. *arXiv preprint arXiv:2206.03352*.

Mavromatis, C.; Subramanyam, P. L.; Ioannidis, V. N.; Adeshina, S.; Howard, P. R.; Grinberg, T.; Hakim, N.; and Karypis, G. 2021. TempoQR: Temporal Question Reasoning over Knowledge Graphs. In *AAAI*.

Nt, H.; and Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Peters, M. E.; Neumann, M.; Logan, R.; Schwartz, R.; Joshi, V.; Singh, S.; and Smith, N. A. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. J.; et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 3982–3992.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kočiskỳ, T.; and Blunsom, P. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Saxena, A.; Chakrabarti, S.; and Talukdar, P. 2021. Question Answering Over Temporal Knowledge Graphs. In *Proceedings of the Annual Meeting of the ACL and IJCNLP*, 6663–6676.

Saxena, A.; Tripathi, A.; and Talukdar, P. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 4498–4507.

Shang, C.; Wang, G.; Qi, P.; and Huang, J. 2022a. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8017–8026. Dublin, Ireland: Association for Computational Linguistics.

Shang, C.; Wang, G.; Qi, P.; and Huang, J. 2022b. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. In *ACL*.

Sharma, A.; Saxena, A.; Gupta, C.; Kazemi, S. M.; Talukdar, P.; and Chakrabarti, S. 2022. TwiRGCN: Temporally Weighted Graph Convolution for Question Answering over Temporal Knowledge Graphs. *arXiv preprint arXiv:2210.06281*.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3): 83–98.

Song, J.; Liang, D.; Li, R.; Li, Y.; Wang, S.; Peng, M.; Wu, W.; and Yu, Y. 2022. Improving Semantic Matching through Dependency-Enhanced Pre-trained Model with Adaptive Fusion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 45–57. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Tan, C.; Wei, F.; Wang, W.; Lv, W.; and Zhou, M. 2018. Multiway Attention Networks for Modeling Sentence Pairs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization.

Trouillon, T.; Dance, C. R.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2017. Knowledge graph completion via complex tensor factorization. *arXiv preprint arXiv:1702.06879*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, S.; Liang, D.; Song, J.; Li, Y.; and Wu, W. 2022. DABERT: Dual Attention Enhanced BERT for Semantic Matching. In *Proceedings of the 29th International Conference on Computational Linguistics*, 1645–1654. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Xian, Y.; Fu, Z.; Muthukrishnan, S.; De Melo, G.; and Zhang, Y. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 285–294.

Xue, C.; Liang, D.; Wang, S.; Zhang, J.; and Wu, W. 2023. Dual Path Modeling for Semantic Matching by Perceiving Subtle Conflicts. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Yasunaga, M.; Bosselut, A.; Ren, H.; Zhang, X.; Manning, C. D.; Liang, P. S.; and Leskovec, J. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35: 37309–37323.

Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; and Leskovec, J. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 535–546. Online: Association for Computational Linguistics.

Zhang, Y.; Dai, H.; Kozareva, Z.; Smola, A.; and Song, L. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Zheng, R.; Rong, B.; Zhou, Y.; Liang, D.; Wang, S.; Wu, W.; Gui, T.; Zhang, Q.; and Huang, X. 2022. Robust Lottery Tickets for Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2211–2224. Dublin, Ireland: Association for Computational Linguistics.

Zhu, Z.; Galkin, M.; Zhang, Z.; and Tang, J. 2022. Neural-symbolic models for logical queries on knowledge graphs. In *International Conference on Machine Learning*, 27454–27478. PMLR.