

Do as I can, not as I get: Topology-aware multi-hop reasoning on multi-modal knowledge graphs

Shangfei Zheng, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Wei Chen, and Lei Zhao

Abstract—Multi-modal knowledge graph (MKG) includes triplets that consist of entities and relations and multi-modal auxiliary data. In recent years, multi-hop multi-modal knowledge graph reasoning (MMKGR) based on reinforcement learning (RL) has received extensive attention because it addresses the intrinsic incompleteness of MKG in an interpretable manner. However, its performance is limited by empirically designed rewards and sparse relations. In addition, this method has been designed for the transductive setting where test entities have been seen during training, and it works poorly in the inductive setting where test entities do not appear in the training set. To overcome these issues, we propose **TMR** (**T**opology-aware **M**ulti-hop **R**easoning), which can conduct MKG reasoning under inductive and transductive settings. Specifically, TMR mainly consists of two components. (1) The topology-aware inductive representation captures information from the directed relations of unseen entities, and aggregates query-related topology features in an attentive manner to generate the fine-grained entity-independent features. (2) After completing multi-modal feature fusion, the relation-augment adaptive RL conducts multi-hop reasoning by eliminating manual rewards and dynamically adding actions. Finally, we construct new MKG datasets with different scales for inductive reasoning evaluation. Experimental results demonstrate that TMR outperforms state-of-the-art MKGR methods under both inductive and transductive settings.

Index Terms—Multi-hop reasoning, multi-modal knowledge graphs, inductive setting, adaptive reinforcement learning

1 INTRODUCTION

KNOWLEDGE graphs (KGs) store and manage huge amounts of data in reality and have been widely used in applications, including recommendation systems [43], information retrieval [28], and knowledge question answering [16]. A traditional KG consists of structural triplets that involve entities and relations, such as *(James Cameron, Role_create, RoseBukater)*. In recent years, as multi-modal data has received widespread attention in the field of data science and artificial intelligence, multi-modal knowledge graphs (MKGs) have emerged [27], [45]. As shown in Figure 1(a), an MKG contains extra multi-modal auxiliary data (images and text description) based on structural triplets, which provides diverse modalities of knowledge. However, the intrinsic incompleteness of MKGs severely limits knowledge applications [34]. To address this problem, the multi-modal knowledge graph reasoning (MKGR) technique is proposed to infer missing triplets in MKGs [53]. For instance, given a triple query *(James Cameron, Writer, ?)*, MKGR can utilize both structural and multi-modal auxiliary data to infer the missing entity *Titanic*.

In the literature, existing MKGR methods can be categorized into two types: single-hop reasoning and multi-hop reasoning [53]. The former focuses on modeling score functions for one-step relations that contain relatively less information [34], [45], while the latter represents the latest work that interpretably infers missing elements by combining multi-hop relations and fusing the corresponding multi-modal features [53]. As shown in Figure 1(b), by connecting *(James Cameron, Role_create, Jack Dawson)* and *(Jack Dawson, Hero, Titanic)*, a missing triplet *(James Cameron, Writer, Titanic)* can be inferred. MMKGR [53] stands out as the unique multi-hop MKGR model in existing ones, garnering significant attention for its state-of-the-art (SOTA) performance and interpretability. To effectively utilize both structural features and corresponding multi-modal features, MMKGR first uses a unified gate-attention network to generate multi-modal complementary features with sufficient attention interactions and less noise. Then, these features are fed into a novel complementary feature-aware reinforcement learning (RL) framework. This framework selects a sequence of actions (i.e., multi-hop reasoning paths) to accumulate rewards on the basis of a manually designed 3D reward function. Finally, MMKGR aims to maximize reward values by successfully inferring missing entities and outputs interpretable reasoning paths.

Although MMKGR demonstrates impressive reasoning performance and interpretability on MKGs, there is still scope for enhancing its action and reward design. (1) *3D reward function is limited by manual design*. It comprises three manual sub-rewards, relying on the experience of domain experts and existing data distributions [12]. However, this necessitates time-consuming redesign when adapting to

- H. Yin and L. Zhao are the corresponding authors of this paper.
- S. Zheng, W. Chen, and L. Zhao are with the Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou 215006, China. E-mail: sfzhengsuda@stu.suda.edu.cn, {robertchen, zhaol}@suda.edu.cn.
- H. Yin and T. Chen are with the School of ITEE, The University of Queensland, St Lucia, QLD 4072, Australia. E-mail: db.hongzhi@gmail.com, tong.chen@uq.edu.au.
- Q. Nguyen is with the School of Information and Communication Technology, Griffith University Gold Coast, QLD 4215, Australia E-mail: henry.nguyen@griffith.edu.au.

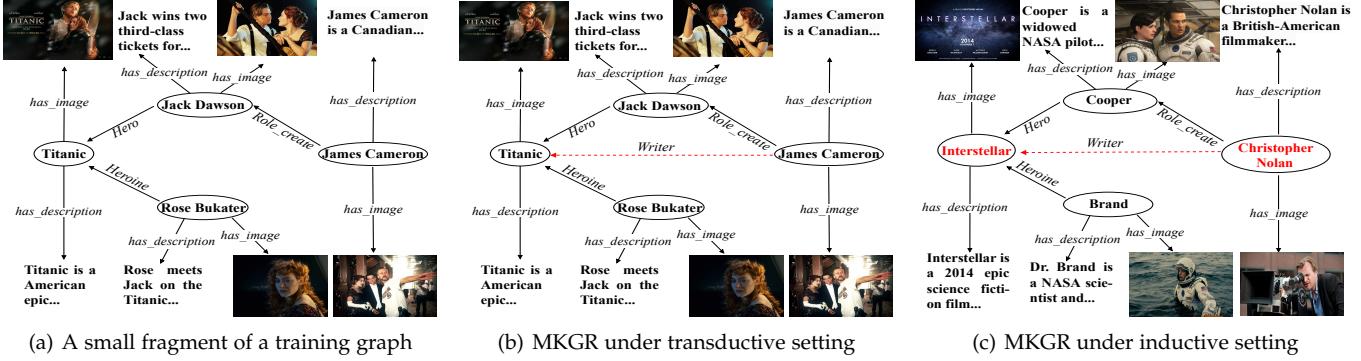


Fig. 1. MKGR under transductive and inductive settings. In the transductive setting, test entities have been seen in the training graph. In contrast, testing entities such as *Interstellar* and *Christopher Nolan* did not appear in the training graph under the inductive setting.

new environments [2]. Moreover, the subjective nature of manual reward design can lead to variations among different designers [22]. (2) *MMKGR is sensitive to the sparsity of relations*. The selection of actions in MMKGR relies on the combination of multi-hop relations. The absence of any relation in this path causes the reasoning path to be unavailable, which limits the reasoning performance [50], [52]. For example, MMKGR infers the missing entity *Titanic* through a two-hop reasoning path $\text{James Cameron} \xrightarrow{\text{Role_create}} \text{Jack Dawson} \xrightarrow{\text{Hero}} \text{Titanic}$. If *Role_create* or *Hero* is unconnected, the aforementioned two-hop path does not exist, which results in the query (*James Cameron*, *Director*, ?) cannot be inferred. Arguably, it is extremely challenging to design an adaptive reward without manual intervention and dynamically add actions to alleviate sparsity.

More importantly, MMKGR is difficult to apply to real scenarios since it primarily concentrates on the transductive setting while overlooking the importance of the inductive setting. As shown in Figure 1 (b), all entities are assumed to be seen during testing in the transductive setting [37]. However, knowledge is evolving and new entities are constantly emerging in the real world [29]. This observation is more in line with the inductive setting where the inferred entities in the test set do not appear in the training set [41]. These inferred entities, often referred to as unseen entities, lack knowable structural representations under the inductive setting. Intuitively, MMKGR can leverage multi-modal auxiliary data of unseen entities to infer the missing triples associated with them. This naturally raises an intriguing and fundamental question: How does MMKGR perform under the inductive setting? To answer this question, we first construct datasets of the induction setting where the entities in the test set and the train set are disjoint [33]. Then, we apply MMKGR to conduct multi-hop reasoning under the inductive setting. Experimental results reveal that MMKGR struggles to converge and has low reasoning performance under the inductive setting. Actually, the conclusion of inductive reasoning methods [33], [47] on traditional KGs is consistent with the above experimental findings: a transductive reasoning method that relies only on multi-modal auxiliary data and lacks generalizability to unseen entities is unsuitable for the inductive setting [11], [25], [41]. This prompts a subsequent goal: *How to develop the inductive capability of MMKGR to generalize unseen entities under an inductive setting?*

A technical challenge to achieving this goal lies in the lack of fine-grained entity-independent representations in existing MKGR methods. One of the key advantages of learning this representation is the development of the inductive capability to generalize unseen entities even in the absence of their specific structural features. [25], [37]. MMKGR lacking the inductive capability has no choice but to use multi-modal auxiliary features of unseen entities to understand these entities, which is highly dependent on the quality and quantity of multi-modal data and not suitable for unseen tasks [11], [41]. Additionally, existing entity-independent representation methods of inductive reasoning on traditional knowledge graph reasoning cannot be directly extended in MMKGR. This is because these methods struggle to aggregate the most relevant information based on specific query relations, resulting in the generation of coarse-grained representations for unseen entities. To make matters worse, the coarse-grained representation of each entity in the reasoning path iteratively disrupts decision-making abilities, which impairs the reasoning performance of MMKGR. Consequently, the fine-grained entity-independent representation is crucial to develop an inductive capability for MMKGR.

In light of the aforementioned challenges in MMKGR, we propose an extended method entitled **TRM** (Topology-aware Multi-hop Reasoning). The main difference between our method and existing ones is that TRM has not only a talent for exploiting multi-modal data, but also the inductive capability to generalize unseen entities. Thus, TRM is capable of conducting MKGR under both inductive and transductive settings. Specifically, TRM mainly contains *topology-aware inductive representation* (TAIR) and *relation-augment adaptive reinforcement learning* (RARL). To develop the inductive capability for MMKGR, TAIR learns fine-grained entity-independent representation from query-related topology knowledge. Its relation-aware entity initializer captures a coarse-grained entity-independent representation by leveraging type information of unseen entities from the connected directed relations. To further generate the fine-grained representation, an adaptive topology representation module introduces a query-aware graph neural network (GNN) to attentively capture the topological information. After completing multi-modal feature fusion, RARL infers the missing elements by multi-hop reasoning path on MKGs, aiming to further improve the comple-

mentary feature-aware reinforcement learning framework of MMKGR. Technically, RARL not only dynamically adds relations as additional actions to eliminate relational sparsity but also adaptively generates rewards by imitating expert demonstrations with filtering low-contributing paths. In summary, as an extension of our conference paper [53], this work makes the following contributions:

- To the best of our knowledge, this is the first work to investigate *how to conduct MKGR under both inductive and transductive settings*.
- To resolve the above problem, we propose an RL-based MKGR model called TMR that mainly contains two components TAIR and RARL. Specifically, TAIR generates fine-grained entity-independent representations to generalize unseen entities. RARL conducts multi-hop reasoning by expanding action space and utilizing imitation learning to eliminate manually designed rewards.
- We construct MKG datasets under the inductive setting. To simulate unseen entities, we ensure that the entities in the test set and training set are disjoint.
- Extensive experiments are conducted under both the transductive and inductive settings. Experimental results demonstrate the superior performance of TMR surpasses MMKGR and various baselines.

The remaining sections are organized as follows. Preliminaries and definitions are presented in Section 2, followed by the overview of TMR. For different components in our proposed model, we introduce them in Sections 4, 5, and 6, respectively. Extensive experiments are shown in Section 7. Section 8 provides a review of the related literature. Finally, we conclude this work in Section 9.

2 PRELIMINARIES AND DEFINITIONS

A MKG is an extension of KG by adding multi-modal auxiliary data, it is denoted as $\mathcal{G}_m = \{\mathcal{E}_m, \mathcal{R}, \mathcal{U}\}$, where \mathcal{R} is a set of semantic relations, and \mathcal{E}_m denotes a set of entities associated with related multi-modal auxiliary data. The features of an entity i are denoted $f_i = f_s \circ f_m$, where “ \circ ” represents a multi-modal fusion method, f_s and f_m denote structural features and multi-modal auxiliary features, respectively. $\mathcal{U} = \{(e_s, r, e_d) \mid e_s, e_d \in \mathcal{E}_m, r \in \mathcal{R}\}$ is a set of triplets, where e_s , e_d , and r denote a head entity, a tail entity, and the relation between these entities, respectively. MKGR typically refers to the link prediction task of the inferring triple query $(e_s, r_q, ?)$ and $(?, r_q, e_d)$, where r_q is a query relation. By adding inverse relation, each triplet (e_s, r, e_d) is equivalent to the triplet (e_d, r^{-1}, e_s) . Without loss of generality, MKGR methods can predict missing head entities by converting $(?, r_q, e_d)$ to $(e_s, r_q^{-1}, ?)$.

Definition 1. *MKG under the transductive setting.* Given a MKG $\mathcal{G}_m = \{\mathcal{E}_m, \mathcal{R}, \mathcal{U}\}$, MKGR under transductive setting aims to reason out a set of triple queries \mathcal{Q} , $\mathcal{Q} = \{(e_s, r_q, ?) \mid (e_s, r_q, ?) \notin \mathcal{U}, e_s, "?" \in \mathcal{E}_m, r_q \in \mathcal{R}\}$, where “?” is a missing entity, $\mathcal{E}_m \in \mathcal{G}_m$ and $\mathcal{R} \in \mathcal{G}_m$ represent entity and relation have been seen in the existing MKG \mathcal{G}_m .

Definition 2. *MKG under the inductive setting.* Given two disconnected MKGs $\mathcal{G}_m = \{\mathcal{E}_m, \mathcal{R}, \mathcal{U}\}$ and $\mathcal{G}_m^* = \{\mathcal{E}_m^*$

$= \{\mathcal{E}_m^*, \mathcal{R}^*, \mathcal{U}^*\}$, \mathcal{G}_m is often known as a training graph, while \mathcal{G}_m^* is considered as a testing graph composed of the triplets by the emerging entities \mathcal{E}_m^* and the relations \mathcal{R}^* . MKGR under the inductive setting requires the model to learn inductive capability in the training graph to infer a set of queries \mathcal{Q} on the test graph, $\mathcal{Q} = \{(e_s, r_q, ?) \mid (e_s, r_q, ?) \notin \{\mathcal{U} \cup \mathcal{U}^*\}, e_s, "?" \in \mathcal{E}_m^*, r_q \in \mathcal{R}^*\}$, where $\mathcal{E}_m \cap \mathcal{E}_m^* = \emptyset$ and $\mathcal{R} \cup \mathcal{R}^* = \mathcal{R}$.

Definition 3. *Multi-hop reasoning.* Multi-hop reasoning infers the missing element by the relational path shorter or equal L hops, where L is an integer not less than 1. A reasoning path is denoted as P , which is obtained by summing all relations and entities in this path.

3 OVERVIEW OF TRM

MMKGR, the version of our conference [53], is limited by manually designed reward functions and relation sparsity as well as poor performance under inductive settings, which motivates us to propose TMR in this paper. As shown in Figure 2, TMR mainly contains two components: **TAIR** and **RARL**. Specifically, inspired by the human inductive ability to generalize unseen tasks from existing relevant knowledge [19], **TAIR** generates fine-grained entity-independent features from the existing topological structure in an attentive manner to represent unseen entities. After employing the unified gate-attention network in MMKGR to complete multi-modal feature fusion, **RARL** conducts MKGR by dynamically adding actions and automatically generating rewards, which is mainly inspired by the fact that humans learn optimal policies by imitating demonstrations rather than predefined paradigms [18]. Notably, TMR is qualified for conducting MKGR under inductive and transductive settings. This is because TMR decouples representation and reasoning into independent components to ensure the flexibility of reasoning under different settings. When inductive settings are converted to transductive setting, TMR only needs to add additional structural representations of seen entities into the unified gate-attention network, while the reasoning module RARL continues to complete reasoning as a multi-modal perception interface without further changes.

4 TOPOLOGY-AWARE INDUCTIVE REPRESENTATION

Existing methods are powerless to capture fine-grained entity-independent representation, thereby restricting the inductive capability of MKGR models [11], [25], [41]. To address the problem, we propose a novel representation method called TAIR in this section. Notably, the technical difference from the existing method for representing unseen entities lies in the following two points: (1) Taking full advantage of type information derived from the connected directed relations of unseen entities. (2) Aggregating query-related neighbor relations in an attentive manner. Specifically, TAIR includes two modules, i.e., a relation-aware entity initializer and an adaptive topology representation. The former obtains coarse-grained representations of the unseen entity, and the latter aggregates topology information related to the query to generate fine-grained entity-independent representations.

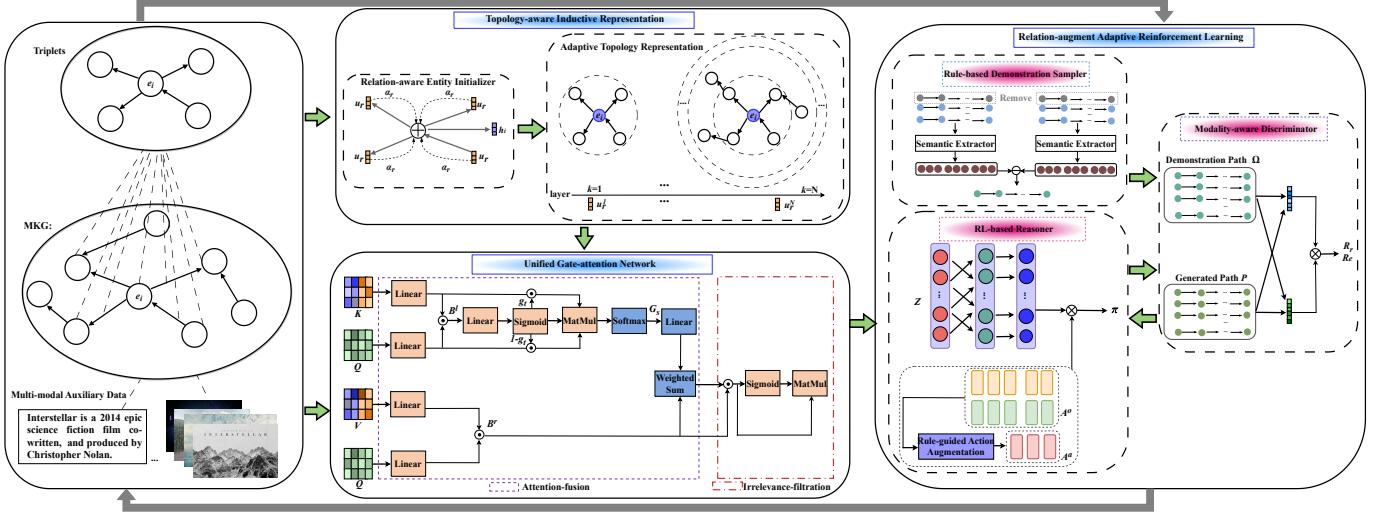


Fig. 2. TAIR first exploits query-related topological information to obtain fine-grained entity-independent features. Then, these features and multi-modal auxiliary features are fed into the UGAN to generate multi-modal complementary features Z . Next, the RL-based reasoner utilizes Z and the augmented actions to generate reasoning paths. The discriminator compares reasoning paths and demonstrations to output adaptive rewards for the reasoner. Finally, the reasoner updates the reasoning policies and interacts with MKG to complete the prediction.

4.1 Relation-aware Entity Initializer

In general, entities with similar semantics have similar topological structures in MKGs, which are reflected in the connection patterns of their incoming and outgoing relations. By analyzing the connection patterns, we can obtain a coarse-grained representation that includes type information for unseen entities. For example, unseen entities *James_Cameron* and *Christopher_Nolan* both contain the outgoing edge *Role_create*, and these two entities have the same type-level representation, i.e., *art creator*.

For an unseen entity e_i , its initialized embedding $\mathbf{h}_i^0 \in \mathbb{R}^d$ as follows:

$$\mathbf{h}_i^0 = \frac{\sum_{r \in I(i)} \mathbf{W}_i \mathbf{u}_r + \sum_{r \in O(i)} \mathbf{W}_o \mathbf{u}_r}{|I(i)| + |O(i)|} \quad (1)$$

where $\mathbf{W}_i, \mathbf{W}_o \in \mathbb{R}^{d \times d}$ are transformation matrices. $I(i)$ and $O(i)$ represent the set of incoming and outgoing relations of the entity e_i , respectively. $\mathbf{u}_r \in \mathbb{R}^d$ is embedding of the relation r .

Considering that the semantics of the same entity can be diverse under different query relations [39], we utilize the attention mechanism to filter out irrelevant neighbor relations. For example, the relations connected by unseen entity *Stephen Curry* have different types, such as family and vocational relations. Given a triple query (*Stephen Curry*, *Father*, ?), the vocational relations connected with the unseen entity indicate that *Stephen Curry* is a professional basketball player, but this information is irrelevant to the query. Therefore, an attention mechanism that dynamically adjusts weights is employed to more accurately represent unseen entities in query tasks. The calculation process is as follows.

$$\alpha_r = \text{softmax}(\mathbf{u}_r, \mathbf{u}_{r_q}) = \frac{\exp(\mathbf{u}_r^\top \mathbf{u}_{r_q})}{\sum_{r' \in \mathcal{N}^r(i)} \exp(\mathbf{u}_{r'}^\top \mathbf{u}_{r_q})} \quad (2)$$

where \mathbf{u}_r and \mathbf{u}_{r_q} are relation representations of neighbor relation r and query relation r_q . α_r denotes the correlation between r and r_q .

After integrating α_r , Eq. (1) is updated as,

$$\mathbf{h}_i^0 = \frac{\sum_{r \in I(i)} \alpha_r \mathbf{W}_i \mathbf{u}_r + \sum_{r \in O(i)} \alpha_r \mathbf{W}_o \mathbf{u}_r}{|I(i)| + |O(i)|} \quad (3)$$

4.2 Adaptive Topology Representation

After obtaining the coarse-grained representation \mathbf{h}_i^0 of the unseen entity e_i by the initializer, we further capture the fine-grained semantic information from the topology of unseen entities. Inspired by the ability of GNNs to capture topology information in knowledge graphs [51], an adaptive topology representation module leverages GNNs to aggregate local structural information from multi-hop neighbors of entity e_i . Specifically, we first concatenate the entities and their relations to obtain triplet information. Compared with individual entities or relations, triple information can provide sufficient topological information [51]. Then, we compute the correlation between the query relation r_q and these triplets that contain more contextual information, which effectively captures the fine-grained representation between topology and r_q [13], [54]. Next, we define updating process of the unseen entity e_i in a k -th layer as follows.

$$\begin{aligned} \mathbf{h}_i^k &= \tanh(\mathbf{W}_{self}^{k-1} \mathbf{h}_i^{k-1} + \sum_{(i', r) \in \mathcal{N}_i(e_i)} \alpha_{i,r} \mathbf{W}_{in}^{k-1} (\mathbf{h}_i^{k-1} \circ \mathbf{u}_r^{k-1}) \\ &\quad + \sum_{(r, i') \in \mathcal{N}_o(e_i)} \alpha_{i,r} \mathbf{W}_{out}^{k-1} (\mathbf{h}_i^{k-1} \circ \mathbf{u}_r^{k-1})) \end{aligned} \quad (4)$$

$$\alpha_{i,r} = \sigma(\mathbf{W}_2 \mathbf{c}_{i,r} + \mathbf{b}) \quad (5)$$

$$\mathbf{c}_{i,r} = \sigma(\mathbf{W}_1 [\mathbf{h}_i^{k-1} \oplus \mathbf{h}_j^{k-1} \oplus \mathbf{u}_r^{k-1} \oplus \mathbf{u}_{r_q}^{k-1}]) \quad (6)$$

where \mathcal{N}_i and \mathcal{N}_o are the incoming and outgoing neighbors of entity e_i , respectively. $\mathbf{W}_{self}^{k-1}, \mathbf{W}_{in}^{k-1}$ and $\mathbf{W}_{out}^{k-1} \in \mathbb{R}^{d \times d}$ denote the transformation matrices, respectively. \circ is the element-wise product, and σ is the activation function *sigmoid*. $\alpha_{i,r}$ is the attention weight of the triplet (e_i, r, e_j) . Based on this, we obtain the fine-grained entity-independent representation \mathbf{h}_i^k of the unseen entity e_i .

Finally, to maintain the consistency of entities and relations within the embedding space, the embeddings of these relations are updated as follows:

$$\mathbf{u}_r^k = \mathbf{W}_r^{k-1} \mathbf{u}_r^{k-1} \quad (7)$$

where $\mathbf{W}_r \in \mathbb{R}^d$ is a transformation matrix.

5 UNIFIED GATE-ATTENTION NETWORK

In this section, we employ the unified gate-attention network (UGAN) in MMKGR to conduct feature fusion of fine-grained entity-independent representation and multi-modal auxiliary representation. Specifically, the unified gate-attention network includes an attention-fusion module, and an irrelevance-filtration module. After extracting multi-modal auxiliary features, the attention-fusion module fuses these features and context features together, by attending them with a carefully designed fine-grained attention scheme. Then, the irrelevance-filtration module discards irrelevant or even misleading information and generates noise robust multi-modal complementary features. Based on this, the unified gate-attention network selects features of different modalities online and simultaneously completes intra-modal and inter-modal attention interactions with noise robustness.

5.1 Feature Extraction

(1) Context features: The entity e_l at reasoning step l and query relation r_q are represented as the fine-grained entity-independent embedding \mathbf{h}_l and \mathbf{u}_{r_q} , respectively. In addition, the history of the reasoning path that consists of the visited entities and relations is defined as $b_l = (e_s, r_0, e_1, r_1, \dots, e_l)$. We leverage LSTM to integrate the vector of history information \mathbf{h}_l with d_s dimensions into context features. Given the query in our multi-hop reasoning process, we obtain the context features $\mathbf{y} = [\mathbf{b}_l; \mathbf{h}_l; \mathbf{u}_{r_q}]$ by concatenating these features. Following [53], a group of context features Y is calculated as $Y = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m]$, where $Y \in \mathbb{R}^{m \times d_y}$, m and d_y are the number of entities and the dimension of the features, respectively. (2) Multi-modal auxiliary features: To initialize image features \mathbf{f}_i , we extract a d_i -dimensional vector of the last fully-connected layer before the softmax in VGG model [4]. Textual features \mathbf{f}_t are initialized by the word2vec framework [30] and expressed as a d_t -dimensional vector. We concatenate the above two parts of features on rows to form the multi-modal auxiliary features $\mathbf{x} = [\mathbf{f}_t W_t; \mathbf{f}_i W_i]$. To flexibly add multi-modal auxiliary features, a group of \mathbf{x} is denoted as $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$, where $W_t \in \mathbb{R}^{d_t \times d_x/2}$, $W_i \in \mathbb{R}^{d_i \times d_x/2}$, and $X \in \mathbb{R}^{m \times d_x}$ represents a group of multi-modal auxiliary features, d_x is the dimension of the feature.

5.2 Attention-fusion Module

To obtain the complementary features with sufficient interactions and less noise, we need to fuse the context features Y and multi-modal auxiliary features X generated in feature extraction. However, redundant features tend to have a negative impact on the prediction during the multi-modal fusion [49]. Specifically, redundant features are either shifted versions of the features related to the triple query or very

similar with little or no variations, which can amplify the negative effects of noise [23]. The redundant features add computational complexity and cause collinearity problems [36]. Consequently, we propose the attention-fusion module that fuses the context features and multi-modal auxiliary features effectively.

Specifically, we first utilize linear functions to generate the queries Q , keys K , and values V of the attention mechanism,

$$Q = XW_q, K = YW_k, V = YW_v \quad (8)$$

where $W_q \in \mathbb{R}^{d_x \times d}$, $W_k, W_v \in \mathbb{R}^{d_y \times d}$, and $Q, K, V \in \mathbb{R}^{m \times d}$ have the same shape. Then, the joint representation B^l of Q and K is learned based on the MLB pooling method [17], inspired by the recent successes of it in fine-grained multi-modal fusion,

$$B^l = KW_k^l \odot QW_q^l \quad (9)$$

Similarly, we can generate the joint representation B^r of V and Q with the following equation,

$$B^r = VW_v^r \odot QW_q^r \quad (10)$$

where $W_k^l, W_q^l, W_v^r, W_q^r \in \mathbb{R}^{d \times j}$ are embedding matrices, and \odot is Hadamard product.

Next, the filtration gate g_t applied to different feature vectors is defined as,

$$g_t = \sigma(B^l W_m) \quad (11)$$

where $W_m \in \mathbb{R}^{j \times d}$ is an embedding matrix and σ denotes the sigmoid activation. Based on the filtration gate g_t , we can filter out the redundant features generated during fusion and obtain a new representation with the following probability distributions,

$$G_s = \text{softmax}((g_t \odot K)((1 - g_t) \odot Q)) \quad (12)$$

where g_t and $1 - g_t$ are used to trade off how many context features and multi-modal auxiliary features are fused.

Finally, our attention-fusion module generates the attended features $\hat{V} = \{\mathbf{v}_i\}_{i=1}^m$ by accumulating the enhanced bilinear values of context features and multi-modal auxiliary features,

$$\hat{V} = \sum_{i=1}^m (G_s W_g^l) B_i^r \quad (13)$$

where $W_g^l \in \mathbb{R}^{d \times 1}$, and $\mathbf{v}_i \in \mathbb{R}^{1 \times j}$ denotes a row of the attended features $\hat{V} \in \mathbb{R}^{m \times j}$, feature vector $B_i^r \in \mathbb{R}^{1 \times j}$ is a row of the embedding matrix B^r .

By designing the attention-fusion module, we can complete the intra-modal and inter-modal feature interactions in a unified manner at the same time. This is because the inputs of this module are pairs from context features and multi-modal auxiliary features, where each vector of a pair may be learned from the same modality or different ones.

5.3 Irrelevance-filtration Module

We use an irrelevance-filtration module to further improve the robustness of the model. The attended features \hat{V} obtained by the attention-fusion module may contain irrelevant features [14]. Specifically, irrelevant features are irrelevant to the triple query in the reasoning process. Since the attention mechanism assigns weights to all features,

these features tend to participate in model computation and mislead the reasoning policy [31]. This motivates our model to weight more on the most related complementary features and dynamically filter irrelevant ones. This is achieved by a well-designed irrelevance-filtration gate function. The output of this gate is a scalar, the value range of which is (0,1). The multi-modal complementary features Z are obtained as follows,

$$G_f = \sigma(B^r \odot \hat{V}) \quad (14)$$

$$Z = G_f(B^r \odot \hat{V}) \quad (15)$$

where σ and G_f denote the sigmoid activation function and irrelevance-filtration gate, respectively.

6 RELATION-AUGMENT ADAPTIVE REINFORCEMENT LEARNING

The existing RL-based MKGR method is limited by manual rewards and sparse relations [1], [20]. To address this problem, we propose a novel RL-based framework entitled RARL in this section. Compared with MMKGR, the main technical difference of RARL lies in the following two points. (1) We effectively increase the additional actions to alleviate the negative impact of sparse relations on the RL-based model. (2) RARL utilizes generative adversarial imitating networks to adaptively learn rewards by imitating demonstrations, which can stabilize reasoning performance and eliminate manual intervention in reward design. This provides a new research perspective for RL-based reasoning methods for MKGR.

RARL consists of three modules namely RL-based reasoner, rule-based demonstration sampler, and modality-aware discriminator. Specifically, the reasoner leverages a rule-guided action augmentation method that dynamically adds additional actions and outputs diverse generated paths about missing elements. Then, the rule-based demonstration sampler filters out low-contributing paths as well as extracts trustworthy demonstrations from MKGs. Next, the modality-aware discriminator generates rewards to update the reasoner by evaluating the semantic similarity between demonstrations and reasoning paths. After sufficient adversarial training, the RL-based reasoner tries to deceive the discriminator to gain more adaptive reward values by imitating the demonstrations. We introduce the above three modules in subsections 6.1, 6.2, and 6.3, respectively.

6.1 RL-based Reasoner

6.1.1 Reinforcement Learning Formulation

RARL trains an agent to interact with the with MKGs by modeling by Markov decision process (MDP). The MDP consists of a 4-tuple, i.e., States, Actions, Transition, Rewards. The agent selects actions based on the current state and obtains rewards from the environment (MKGs) to update its behavior policy until it reaches a termination state or a predefined reasoning step.

States: The state of the agent at reasoning step l is denoted as $s_l = (e_l, (e_s, r_q)) \in \mathcal{S}$, where \mathcal{S} denotes a state space and e_l represents the entity at the current reasoning step l . The source entity e_s and the query relation r_q are the global context shared throughout all steps.

Actions: For the given state s_l , its original action space is the set of usable actions A_l^o at reasoning step l is expressed as $A_l^o = \{(r_{l+1}, e_{l+1}) | (e_l, r_{l+1}, e_{l+1}) \in \mathcal{G}_m\}$. To alleviate relation sparsity, the rule-guided action augmentation module adds extra potential actions A_l^a into the original action space. Thus, the joint action space $A_l^o \cup A_l^a = A_l \in \mathcal{A}$. In addition, we add *STOP* action to avoid infinitely unrolling in the reasoning process. The *STOP* action executes a self-loop when the reasoning step is unrolled to the maximum step L .

Transition: \mathcal{P}_r is defined to facilitate the transition from the current state s_l to the next state s_{l+1} . $\mathcal{P}_r: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined as $\mathcal{P}_r(s_l, A_l) = \mathcal{P}_r(e_l, (e_s, r_q), A^o, A^a)$.

Rewards: Different from existing manually-designed reward functions, we design an adaptive reward mechanism to eliminate manual intervention, which achieves high reasoning performance in complex and uncertain environments. The adaptive reward arises from path comparisons between the generator and expert demonstration, and it is defined in Eq. 29.

Policy Network The policy function π is used as a multi-modal perception interface to output the next action with the highest executable probability. For a given state, π selects the promising action with the maximum likelihood, which is defined as,

$$\pi_\theta(a_l|s_l) = \text{softmax}(\mathbf{A}_l(\text{WReLU}(Z))) \quad (16)$$

where $a_l \in A_l$, and A_l can be encoded to \mathbf{A}_l by stacking the representations of existing available actions.

6.1.2 Rule-guided Action Augmentation

The existing RL-based reasoning method assumes sufficient relation paths between entities, and regards these relations and connected tail entities as the next action. However, the intrinsic incompleteness of MKGs leads to sparse relations of an entity. Especially, emerging entities are sparsely connected to existing entities under the inductive setting. This sparsity limits the utilization of potential reasoning paths. Therefore, it is necessary to design an action space augmentation method to eliminate the sparsity of the action space. Although the idea of augmenting the action space is promising, a major challenge is how to *efficiently augment additional actions*.

Intuitively, enumerating all relations and entities to compose an additional action space can complement the existing action space. However, this combined search space is close to $\mathcal{O}(|\mathcal{E}| \times |\mathcal{R}|)$, where $|\mathcal{E}|$ and $|\mathcal{R}|$ are the numbers of entities and relations in a MKG, respectively. For a large-scale MKG with millions of entities and thousands of relations, large search space becomes an intractable problem. To address the above problem, we propose a novel action augmentation method to efficiently augment additional actions. For a state s_l , the candidate set of augmented action is denoted as $C_t = \{(r', e') | r' \in \mathcal{R} \wedge e' \in \mathcal{E}_m \wedge (e_l, r', e') \notin \mathcal{G}_m\}$. First, we calculate the probability for the candidate set C_t :

$$p((r', e')|s_l) = p(r'|s_l)p(e'|r', s_l) \quad (17)$$

To reduce the candidate action space and time-consuming, an approximate pruning strategy is used to filter out additional actions. The pruning strategy consists

of additional relation selection using $p(r'|s_l)$ and entity generation using $p(e'|r', s_l)$.

Then, for the state s_l , the attention score of the candidate relations is calculated as $p(r|s_l)$,

$$\mathbf{w} = \text{softmax}(MLP(\mathbf{s}_l) \cdot [\mathbf{u}_{r_1}, \dots, \mathbf{u}_{r_{|\mathcal{R}|}}]) \quad (18)$$

where \mathbf{w} denotes the attention vector. We select top x relations with the largest attention values in \mathbf{w} to obtain additional relation set $\mathcal{R}_{add} = r^1, r^2, \dots, r^I$.

Next, we leverage AnyBURL [8] to mine logical rules and corresponding confidence scores from the MKGs to guide the generation of tail entities. A logical rule \mathcal{B} is defined as:

$$\wedge_{i=1}^N r_i(e_i, e_{i+1}) \Rightarrow r_h(e_1, e_N) \quad (19)$$

where \wedge is logic conjunction, N denotes the maximum length of the rule chain. The left side is a rule body, which is represented by a conjunction of atoms $(e_1, r_1, e_2) \wedge \dots \wedge (e_i, r_i, e_{i+1}) \wedge \dots \wedge (e_{N-1}, r_{N-1}, e_N)$. In addition, the right-side is called a rule head, r_h denotes the head relation.

We further measure the quality of each \mathcal{B} by a confidence score, it is calculated as:

$$\text{conf}(\mathcal{B}) = \frac{\text{pos}(\mathcal{B})}{\text{pos}(\mathcal{B}) + \text{neg}(\mathcal{B})} \quad (20)$$

where $\text{pos}(\mathcal{B})$ and $\text{neg}(\mathcal{B})$ are the numbers of positive and negative trajectories, respectively.

Finally, we predict the tail entities $p(e'|r', s_l)$ by retrieving the mined rules. For an additional relation r' , we select the mined rules with the highest confidence $\text{conf}(\mathcal{B})$ to select the tail entity, which can complete new facts. The fact set \mathcal{F} can be formulated as,

$$\begin{aligned} \mathcal{F} = & \{(e_l, r', e') | \exists \{(e_l, r_1, e_1), (e_1, r_2, e_2), \dots, (e_n, r_n, e') \\ & \in \mathcal{K} \wedge (r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_n) \in \mathcal{B} \wedge r' \in \mathcal{R}\}\} \end{aligned} \quad (21)$$

where \mathcal{K} is a set of atoms. We add additional actions to the multi-hop reasoning process, which effectively eliminates the sparsity of the action space. With this implementation, our overall search space of action augmentation is $\mathcal{O}(|\mathcal{R}| + |\mathcal{I}| \times |\mathcal{K}|)$, where \mathcal{I} and $\mathcal{K} \ll \mathcal{R}$.

6.2 Rule-based Demonstration Sampler

Inspired by the fact that learning reasoning rules can increase the trustworthiness of the demonstrations [15], we extract a set of rule paths P_{r_q} of the query relation r_q as candidate demonstrations. Then, we reserve a set of the shortest paths P'_{r_q} with size N to filtrate noisy paths. This is because the combination of the shortest relation path can accurately represent the semantic link between two entities [7], [46]. And longer paths are more likely to contain worthless or even noisy information [24]. Next, we encode the path package μ by concatenating all path embeddings in P'_{r_q} . Inspired by ConvKB showcased the effectiveness of convolutional neural networks (CNNs) in extracting semantic features from paths, the path package denoted as $\mu \in \mathbb{R}^{Nd}$ is fed into a convolutional layer by sliding a kernel ω :

$$\mathbf{c}_1 = W_2 \text{Relu}(W_1 \text{ReLU}(\text{Conv}(\boldsymbol{\mu}, \boldsymbol{\omega}))) \quad (22)$$

To further select the rule paths related to r_q , we design a counterfactual filtering method. We use the dot product to extract the correlation between \mathbf{u}_{r_q} and \mathbf{c}_1 .

$$t_\mu = \sigma(\mathbf{c}_1 \cdot \mathbf{u}_{r_q}) \quad (23)$$

where t_μ is the correlation score to evaluate the path package representation \mathbf{c}_1 and the query relation \mathbf{u}_{r_q} . The counterfactual filtering method removes each rule path x' from the P'_{r_q} in turn. The altered path set is calculated by Eq. 23 to obtain a new correlation score. we refine the demonstration set according to this score:

$$\Omega = \{x | t_u - t_{u-x'} > 0, x \in x'\} \quad (24)$$

where Ω is the refined demonstration containing N' paths. If the path x' is important to the query relation, the removal of x' will lead to the result of $t_u - t_{u-x'}$ being positive.

6.3 Modality-aware Discriminator

Typically, relations in paths mainly focus on connections between entities, while entities focus on multi-modal representations. When evaluating demonstrations and generated paths, dividing entities and relations into two levels not only avoids information confusion but also better adapts to the variation of semantic features at different levels.

Relation Level The discriminator aims to judge whether the demonstrations and generated paths are similar in relational connectivity. For a triple query, a set of reasoning paths P are packaged by concatenating all path embeddings and denoted as ν .

For a triple query, we first remove entities in Ω and P to extract relation paths by summing all relations. By using the concatenation operation to package the path, we obtain relation packages ν_Ω and ν_P . Next, we extract semantic features \mathbf{c}_Ω and \mathbf{c}_P using Eq. 22. Finally, the sigmoid function is used to the features to the interval (0,1).

$$D(\boldsymbol{\nu}_\Omega) = \sigma(W_{s2} \text{ReLU}(W_{s1} \mathbf{c}_\Omega)) \quad (25)$$

where σ is a sigmoid function. We can obtain the representation $D(\boldsymbol{\nu}_\Omega)$ of \mathbf{c}_1 in the same calculation form with $D(\boldsymbol{\nu}_P)$.

Entity Level To better identify the differences between multi-modal features of entities, we leverage multi-modal complementary features in paths to accomplish entity-level discrimination. Specifically, for a triple query, we first gather the entities in the set of paths in Ω and P . Then, we also use the concatenation operation to package the entities to obtain two entity packages κ_Ω and κ_P . Finally, we employ a multi-layer neural network as the feature extractor, which is defined as follows,

$$D(\boldsymbol{\kappa}_\Omega) = \sigma(W_e \tanh(\boldsymbol{\kappa}_\Omega)) \quad (26)$$

where $D(\boldsymbol{\kappa}_\Omega)$ is formulated as the package features of entities. We can obtain $D(\boldsymbol{\kappa}_P)$ n the same calculation form as $D(\boldsymbol{\kappa}_\Omega)$.

6.4 Training

In this section, we define the final reward function and formalize the training process. Specifically, we first introduce

TABLE 1

Statistics of the experimental datasets over the transductive setting.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
WN9-IMG-TXT	6,555	9	11,747	1,337	1,319
FB-IMG-TXT	11,757	1,231	285,850	29,580	34,863

path features whose quality is higher than random noise into the reward.

$$R_r = \max(D(\nu_P) - D^N(\nu_P), 0) \quad (27)$$

$$R_e = \max(D(\kappa_P) - D^N(\kappa_P), 0) \quad (28)$$

where $D^N(\nu_P)$ and $D^N(\kappa_P)$ respectively denote noise embeddings from the entity and relation layers. These embeddings consist of random noise sampled from a continuous uniform distribution. R_r and R_e represent the adaptive rewards from relation layer and entity layer, respectively.

Then, we define the adaptive reward $R(s_l)$ at the state s_l as follows:

$$R(s_l) = \alpha R_e + (1 - \alpha) R_r \quad (29)$$

where α is a balance factor. Note that, the agent learns adaptive rewards from demonstrations without manually designing and tuning, which reduces manual intervention and subjective bias [2]. In addition, the adaptive reward improves the generalizability of our proposed model and it is suitable for unseen tasks. This is because the adaptive reward mechanism automatically capturing common meta-knowledge by learning relation patterns and multi-modal features in demonstrations [32].

Next, we optimize the modality-aware discriminator by reducing training loss and expect it to possess expertise in distinguishing between P and Ω .

$$\mathcal{L}_r = D(\nu_P) - D(\nu_\Omega) + \lambda (\|\nabla_{\hat{p}} D(\hat{p})\|_2 - 1)^2 \quad (30)$$

where λ is a penalty term and \hat{p} is sampled uniformly along straight lines between the generated path and the demonstration.

For the discriminator at the entity level, we define loss \mathcal{L}_e as follows:

$$\mathcal{L}_e = -(\log D(\kappa_\Omega) + \log(1 - D(\kappa_P))) \quad (31)$$

Finally, to maximize the accumulated rewards of adaptive reinforcement learning and obtain the optimal policy, the objective function is as follows,

$$\mathcal{J}(\theta) = \mathbb{E}_{(e_s, r, e_d) \sim \mathcal{G}_m} \mathbb{E}_{a_1, \dots, a_L \sim \pi_\theta} [R(s_l | e_s, r)] \quad (32)$$

7 EXPERIMENT

7.1 Datasets

Following MMKGR [34], [53], we use WN9-IMG-TXT and FB-IMG-TXT to verify the reasoning performance under the transductive setting. Each entity in these MKGs contains three modal information: structure, image, and text. Specifically, the relation triplets and textual descriptions are extracted from WordNet and Freebase. To extract the image features of the entities, 10 images and 100 images are crawled for each entity in WN9-IMG-TXT and FB-IMG-TXT, respectively [34]. Statistics are shown in Table 1.

To perform inductive reasoning in MKGs, we construct new inductive benchmark datasets by extracting disjoint

TABLE 2
Statistics of the datasets over the inductive setting.

WN9-IMG-TXT			FB-IMG-TXT				
	#Ent	#Rel	#Triplets	#Ent	#Rel	#Triplets	
V1	Training	420	8	760	2848	342	17561
V1	Ind_test	270	8	401	2002	263	3325
V2	Training	654	9	1465	3205	631	35184
V2	Ind_test	406	8	814	2111	343	6064
V3	Training	658	9	2180	3716	750	52544
V3	Ind_test	581	8	1442	2419	254	10623

subgraphs from WN9-IMG-TXT and FB-IMG-TXT. In particular, each dataset contains a pair of graphs: *training graph* and *ind_test graph*. Following [37], to generate the training graph, we first uniformly sample several entities as the root nodes, and then conduct the union of k-hop neighbor triplets around the roots. Next, we set the maximum number of samples at each hop to prevent the exponential growth of new neighbors. Finally, we remove the training graph from the whole graph and sample the test graph using the same procedure. In fact, the above division method destroys the link distribution in the original graph and reduces the number of triplets. Therefore, for a robust evaluation, we adjust the parameters of the above procedure to sample 5%, 10%, and 15% of the original graph (i.e., WN9-IMG-TXT and FB-IMG-TXT) to construct datasets with different sizes [41]. In summary, the model is trained on the training graph and tested on the ind_test graph in every version. Note that, (1) the two graphs have disjoint sets of entities, (2) training graphs contain all relations present in ind_test graphs.

7.2 Evaluation Protocol

To evaluate the reasoning performance of TMR over inductive and transductive settings, we adopt the mean reciprocal rank (MRR) and Hits@N to report experimental results, which are common metrics for MKGR [34], [53].

7.3 Baselines and Implementation Details

To investigate the performance of TMR, two categories of methods are compared. 1) Knowledge graph reasoning methods under the induction setting: DRUM [33], CoMPILE [29], MorsE [6], RED-GNN [51]. 2) SOTA MKGR method: MMKGR [53]. Note that, the transductive reasoning methods (i.e., TransE [3] or RLH [38]) on the traditional KG cannot be applied to MKGR. This is because these methods retrain the embedding from scratch whenever a new entity appears [41]. In our training stage, some core settings are as follows. The embedding dimension d_s of relation and history is set to 200, the embedding dimension d_i of the image feature is set to 128 and 4096 on FB-IMG-TXT and WN9-IMG-TXT respectively, and the embedding dimension d_t of textual feature is 1000. The maximum reasoning step L is 3. The number of additional relation I is 3 and 5 on different versions of WN9-IMG-TXT and FB-IMG-TXT, respectively. The size N is set to 5. We employ a 3-layer GNN to obtain the adaptive topology representation. α in Eq.(29) is set to 0.4 and 0.2 on different versions of WN9-IMG-TXT and FB-IMG-TXT, respectively.

TABLE 3
Inductive link prediction results for different versions of MKGs.

	WN9-IMG-TXT										FB-IMG-TXT									
	V1		V2		V3		V1		V2		V3		V1		V2		V3			
MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10			
MMKGR	27.0	25.1	30.2	27.2	25.5	30.6	29.3	26.9	33.3	21.4	19.3	25.3	23.3	21.7	26.5	26.0	23.9	29.1		
DRUM	43.4	40.5	46.0	45.2	41.6	48.7	48.4	45.7	51.0	35.6	32.7	38.2	37.8	34.8	39.7	40.1	38.6	43.2		
CoMPILE	45.2	41.7	47.3	47.0	43.9	50.8	49.1	47.6	53.2	36.9	34.5	39.3	39.8	35.9	40.9	42.3	39.9	44.4		
MorseE	48.2	45.8	52.4	50.3	48.6	53.1	54.2	52.1	56.7	38.3	36.6	41.2	40.7	38.3	43.1	44.2	42.1	46.2		
RED-GNN	<u>51.2</u>	<u>49.3</u>	<u>54.3</u>	<u>53.3</u>	<u>50.2</u>	<u>56.8</u>	<u>56.1</u>	<u>54.8</u>	<u>59.2</u>	<u>40.5</u>	<u>38.4</u>	<u>42.4</u>	<u>43.3</u>	<u>41.1</u>	<u>45.2</u>	<u>46.0</u>	<u>44.2</u>	<u>48.4</u>		
TMR	64.9	62.3	69.1	67.6	64.1	72.0	71.1	69.7	74.8	57.0	54.7	59.4	60.1	57.5	62.7	63.3	60.8	66.2		
Improv.	13.7%	13.0%	14.8%	14.3%	13.9%	15.2%	15.0%	14.9%	15.6%	16.5%	16.3%	17%	16.8%	16.4%	17.5%	17.3%	16.6%	17.8%		

TABLE 4
Transductive link prediction results on different MKGs.

Model	WN9-IMG-TXT			FB-IMG-TXT		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
MMKGR	80.2	73.6	92.8	71.3	65.8	82.6
TMR-TAIR	<u>83.6</u>	<u>76.4</u>	<u>93.2</u>	<u>74.3</u>	<u>67.5</u>	<u>85.4</u>
TMR	<u>86.3</u>	<u>79.7</u>	<u>93.7</u>	<u>76.6</u>	<u>71.4</u>	<u>87.6</u>

7.4 Inductive Reasoning on MKGs

Reasoning performance over the inductive setting is reported in Table 3 (these scores are in percentage), where the competitive results of the baseline are marked by underlining and the highest performance results are highlighted in bold. Specifically, we have the following insightful analysis based on the experimental results in Table 3. (1) The performance of our proposed TMR outperforms all baselines. This is because TMR not only generates fine-grained entity-independent representations for unseen entities, but also utilizes RARL to eliminate the negative impact of manual rewards and sparse relations on reasoning accuracy. (2) The experimental results of MMKGR are the lowest in the different datasets. This is because MMKGR without induction capability cannot obtain structured representations of unseen entities. Therefore, it is not a reasonable solution to only utilize multi-modal auxiliary information without learning inductive capabilities in the induction setting. (3) The rule-based SOTA model DRUM combines relations to infer unseen tasks, but this model is limited by the quality of the rules. (4) Similar to DRUM, CoMPILE, MorseE and RED-GNN are designed to infer unseen tasks in traditional knowledge graphs, but they adapt to conduct reasoning without utilizing multi-modal auxiliary features in the inductive setting. This is because the three models learn the inductive ability from the local graph structure information. In particular, the existing SOTA model RED-GNN remains competitive in all versions of MKGs. This is because RED-GNN recursively encodes multiple relational digraphs with shared edges and preserves the structural patterns at the same time [51]. In short, the key to performance improvement in the induction setting is to consider both induction capability and the ability to exploit multi-modal data. Note that, TMR is the first model to do this in the domain of MKGR.

7.5 Transductive Reasoning on MKGs

In this section, we investigate whether TMR outperforms the SOTA model MMKGR under the transductive setting. To be consistent with MMKGR, we obtain the pre-trained

embeddings of all entities by TransE. This is because unseen entities do not exist under the transductive setting. Specifically, the pre-trained representation e_i of entity i is additionally added into the context features of TMR. Furthermore, a concern is that TMR is fed with pre-trained entity representations do not need entity-independent features generated by the TAIR component that contains topology information in the transductive setting. To eliminate the above concern, we added a variant of TMR-TAIR where the TAIR component is removed (i.e., the context features of TMR do not contain fine-grained entity-independent features).

The experimental results under the transductive setting are shown in Table 4. We have the following observations. (1) The reasoning performance of TMR surpasses that of the SOTA MMKGR on the original MKGs. This demonstrates the flexibility and generalizability of TMR under different settings. (2) The performance of TMR-TAIR is higher than that of MMKGR. This is because the RARL component in TMR-TAIR can eliminate the negative impact of manual rewards and sparse relations on reasoning accuracy. (3) The performance of TMR-TAIR is lower than that of TMR. A reason is that the entity-independent features generated by the TAIR component aggregate multi-hop topology information. The information can provide reasoning context to improve performance under both inductive and transductive settings.

7.6 Ablation Study

The overall goal of the ablation study is to measure the contribution of different components by adding different variants. Figure 3 reports the experimental results of the TMR and its variants. (1) The variant version w/o TAIR only uses multi-modal features of unseen entities where the TAIR component is removed from the TMR. (2) w/o UGAN, in which the unified gate-attention network is removed and basic concatenation operation is added to fuse all features. (3) w/o RARL, a variant version where RARL is removed. To ensure the agent conducts the reasoning, we retain the RL-based reasoner with the original action space and basic

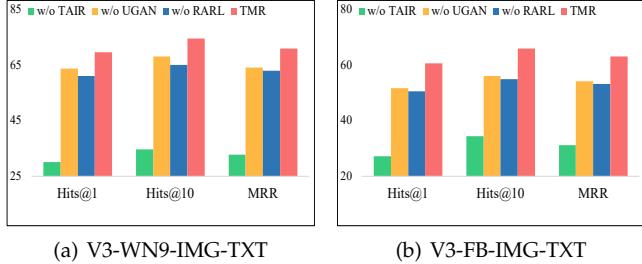


Fig. 3. Ablation on different components of the TMR.

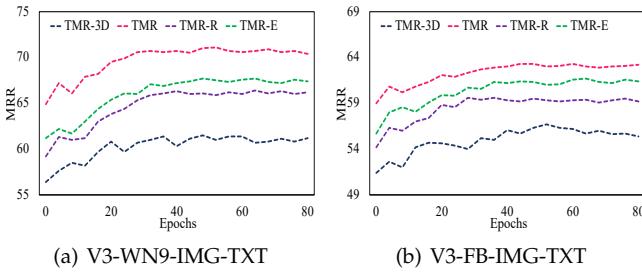


Fig. 4. The convergence rate of TMR and the variant model TMR-3D.

0/1 reward (i.e., the reward value is set to 1 if the target entity is the ground truth entity. Otherwise, the value is 0).

We have the following observations. (1) TMR has the best performance compared with both variant versions. This validates the effectiveness of all components. (2) The results of w/o TAIR are much lower than that of TMR, which verifies the importance of learning fine-grained entity-independent representation in the inductive setting. (3) Although the input of w/o UGAN includes multi-modal information, its reasoning performance still declines compared with TMR. This is because the concatenation operation cannot generate multi-modal complementary features. (4) After removing the rule-guided action augmentation method and adaptive reward mechanism, the performance of w/o RARA significantly degrades on different datasets. This demonstrates that RARL can eliminate the negative effects of sparse relations and manual rewards on reasoning performance.

7.7 Further Analysis

7.7.1 Convergence Analysis

To analyze the convergence rate and reasoning performance between our adaptive reward and manual rewards in MMKGR, we design a variant entitled TMR-3D. Specifically, this variant removes the *Rule-based Demonstration Sampler* and *Modality-aware Discriminator* in RARL. To ensure that the agent is rewarded, we add the 3D reward mechanism (i.e., the manual reward function in MMKGR). In addition, we design variant models TMR-R and TMR-E by removing the relation-level reward R_r and the entity-level reward R_e in TMR, respectively. This setting can investigate the contribution of different parts of the adaptive reward. Observed from Figure 4, (1) TMR adopted the adaptive reward has the fastest convergence and the most stable performance on MKGs. This is because the adaptive reward learned from both path semantic and multi-modal feature automatically eliminate the manual intervention and avoids decision bias

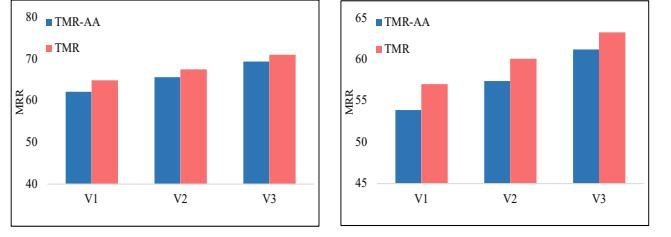
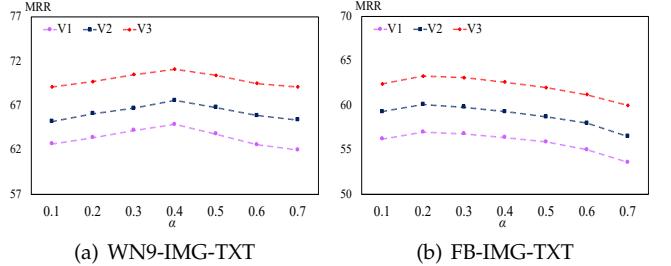


Fig. 5. Performance comparison between TMR and TMR-AA.

Fig. 6. Performance of TMR w.r.t. varied α on different datasets.

on different MKG datasets [21]. (2) Although TMR-3D can converge slowly, its performance is still unstable. A reason is that the weak generalizability of manually-designed 3D rewards leads to unstable training on different datasets [10]. (3) The performance of TMR-R is slightly worse than that of TMR-E, which indicates that R_r has more contribution than R_e in the RARL.

7.7.2 Effectiveness Analysis for Action Augmentation

To investigate the effectiveness of the action augmentation method for expanding the latent action space, we design a new variant model TMR-AA by removing the rule-guided action augmentation method in RARL. The experimental results are shown in Figure 5, and we have the following analysis: (1) The performance of TMR has declined to varying degrees in all datasets after removing the action augmentation method, which verifies the effectiveness of the proposed action augmentation method. (2) The performance improvement of the rule-guided action augmentation method on different versions of FB-IMG-TXT is more obvious than that on the different versions of WN9-IMG-TXT. This is because more relations on different versions of FB-IMG-TXT can be used to build more complex reasoning rules. (3) Compared with adaptive rewards, the performance improvement of the rule-guided action augmentation method is relatively small. One potential reason is that the reasoning processes completely depend on original actions, and the additional actions from this method mainly play an auxiliary role.

7.8 Key Parameter Analysis

The balance factor α is an important parameter for our proposed model TMR. As presented in Figure 6, 0.4 and 0.2 are the optimal values of α on the different versions of WN9-IMG-TXT and FB-IMG-TXT, respectively. The reward R_e from entity level is assigned small weights, which demonstrates that the semantic correctness of relational paths

provides better reasoning clues than multi-modal entities in inductive reasoning tasks.

8 RELATED WORK

8.1 Multi-modal Knowledge Graph

A traditional KG is essentially a semantic graph that consists of entities (nodes) and relations (edges). At present, the actual internet data show multi-modal characteristics [15]. MKGs are developed to incorporate various types of data from different modalities into KGs [34]. A MKG typically includes structural triplets, and multi-modal data (i.e., texts and images) [26]. Common MKGs are IMGpedia [9], Richpedia [40], and FB-Des [36]. However, the multi-modal auxiliary data of these MKGs is singular (i.e., image or text). To expand the auxiliary data with one modality, WN9-IMG-TXT and FB-IMG-TXT simultaneously add a number of textual descriptions and images to each entity, aiming to further enhance the data diversity of the MKGs [34], [48].

8.2 Multi-modal Knowledge Graph Reasoning

Since MKGs inherently contain incomplete knowledge, MKGR technology that can synthesize the original knowledge and infer the missing knowledge is particularly important [34], [35]. Some studies employ the attention model or concatenation to fuse multi-modal auxiliary features and then adopt TransE to infer missing elements, such as IKRL [44] and TransAE [42], and MTRL [34]. However, these methods lack interpretability and are primarily suitable for single-hop reasoning containing limited information. To address this issue, MMKGR leverages the symbolic compositionality of the multi-step relational path (choices of actions) to infer the correct entity [5], [53]. MMKGR has been proven to be a SOTA model in the field of TKGR. Its multi-hop reasoning process is as intuitive as “going for a walk”, which naturally forms an explainable provenance for MMKGR.

8.3 Inductive Reasoning on Knowledge Graph

The inductive setting is receiving increasing attention since unseen entities are emerging in KGs. Therefore, completing knowledge reasoning in an inductive setting is a practical application. Several methods are proposed to solve this problem. Rule-based methods can leverage the logical rules of existing knowledge to infer new facts, because the rules are independent of specific entities [33]. In addition, GraIL [37] and CoMPILE [29] aim to generalize to unseen entities and improve reasoning performance by subgraph extraction, but the enclosing subgraphs cannot learn relational structures so as to weaken the inductive capability. Inspired by the powerful graph modeling capabilities of, SOTA models like MorsE [6] and RED-GNN [51] utilize GNNs to aggregate topological structures and mine existing neighbor information, which is a promising method for inductive reasoning. However, these methods still have limitations: (1) They do not extract fine-grained entity-independent features related to the query, which restricts their inductive capacity. (2) Lack of ability to utilize multi-modal auxiliary information in MKGR.

9 CONCLUSION

In this paper, we propose TMR as a solution to conduct TKGR in both inductive and transductive settings. Specifically, TMR mainly includes TAIR and RARL. TAIR learns fine-grained entity-independent representation from query-related topology knowledge to represent unseen entities. RARL eliminates the negative impact of sparse relations and artificial rewards on reasoning accuracy by introducing additional actions and adaptive rewards. To ensure that the entities in the training and testing sets are disjoint under the inductive setting, we construct six MKG datasets with varying scales. Experimental results demonstrate the superior performance of our proposed model compared to existing baselines across different settings.

REFERENCES

- [1] S. Arora and P. Doshi, “A survey of inverse reinforcement learning: Challenges, methods and progress,” *Artif. Intell.*, pp. 500–512, 2021.
- [2] S. Arora, A. Prashant, and P. Doshi, “A survey of inverse reinforcement learning: Challenges, methods and progress,” *Artif. Intell.*, pp. 103 500–103 511, 2021.
- [3] A. Bordes, N. Usunier, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *NeurIPS*, 2013, pp. 2787–2795.
- [4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *BMVC*, 2014, pp. 1–11.
- [5] H. Chen, Y. Li, X. Sun, G. Xu, and H. Yin, “Temporal meta-path guided explainable recommendation,” in *WSDM*, 2021, pp. 1056–1064.
- [6] M. Chen, W. Zhang, and Y. Zhu, “Meta-knowledge transfer for inductive knowledge graph embedding,” in *SIGIR*, 2022, pp. 927–937.
- [7] X. Chen, S. Jia, and Y. Xiang, “A review: Knowledge reasoning over knowledge graph,” *Expert Syst. Appl.*, pp. 948–962, 2020.
- [8] M. Christian, C. Wudage, and R. Daniel, “Anytime bottom-up rule learning for knowledge graph completion,” in *IJCAI*, 2019, pp. 3137–3143.
- [9] S. Ferrada, B. Bustos, and A. Hogan, “Imgpedia: A linked dataset with content-based analysis of wikipedia images,” in *ISWC*, 2017, pp. 84–93.
- [10] J. Gui, Z. Sun, and Y. Wen, “A review on generative adversarial networks: Algorithms, theory, and applications,” *IEEE Trans. Knowl. Data Eng.*, pp. 3313–3332, 2023.
- [11] Y. He and P. Zhang, “VN network: Embedding newly emerging entities with virtual neighbors,” in *CIKM*, 2020, pp. 505–514.
- [12] Z. Hou, X. Jin, and Z. Li, “Rule-aware reinforcement learning for knowledge graph reasoning,” in *IJCNLP*, 2021, pp. 4687–4692.
- [13] J. Huang and T. Zhang, “A deep embedding model for knowledge graph completion based on attention mechanism,” *Neural Comput. Appl.*, pp. 9751–9760, 2021.
- [14] L. Huang, W. Wang, J. Chen, and X. Wei, “Attention on attention for image captioning,” in *ICCV*, 2019, pp. 4633–4642.
- [15] S. Ji, S. Pan, and E. Cambria, “A survey on knowledge graphs: Representation, acquisition, and applications,” *IEEE Trans. Neural Networks Learn. Syst.*, pp. 494–514, 2022.
- [16] M. Kaiser, R. S. Roy, and G. Weikum, “Reinforcement learning from reformulations in conversational question answering over knowledge graphs,” in *SIGIR*, 2021, pp. 459–469.
- [17] J. Kim, K. On, W. Lim, and B. Zhang, “Hadamard product for low-rank bilinear pooling,” in *ICLR*, 2017, pp. 1–14.
- [18] J. Kober and J. Peters, “Imitation and reinforcement learning,” *IEEE Robotics Autom. Mag.*, pp. 55–62, 2010.
- [19] S. Kumar, C. Correa, and I. Dasgupta, “Using natural language and program abstractions to instill human inductive biases in machines,” in *NeurIPS*, 2022, pp. 167–180.
- [20] P. Ladosz, L. Weng, and M. Kim, “Exploration in deep reinforcement learning: A survey,” *Inf. Fusion*, pp. 1–22, 2022.
- [21] A. Li and P. Abbeel, “Generalized hindsight for reinforcement learning,” in *NeurIPS*, 2020.

- [22] A. Li, L. Pinto, and P. Abbeel, "Generalized hindsight for reinforcement learning," in *NeurIPS*, 2020, pp. 7754–7767.
- [23] P. Li, H. Liu, Y. Si, C. Li, and F. Li, "EEG based emotion recognition by combining functional connectivity network and local activations," *IEEE Trans. Biomed. Eng.*, pp. 2869–2881, 2019.
- [24] X. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," in *EMNLP*, 2018, pp. 3243–3253.
- [25] S. Liu, B. Grau, and I. Horrocks, "INDIGO: gnn-based inductive knowledge graph completion using pair-wise encoding," in *NeurIPS*, 2021, pp. 2034–2045.
- [26] Y. Liu, H. Li, and D. Rosenblum, "MMKG: multi-modal knowledge graphs," in *EWSWC*, 2019, pp. 459–474.
- [27] Y. Liu, H. Li, and D. S. Rosenblum, "Mmkgr: Multi-modal knowledge graphs," in *EWSWC*, vol. 11503, 2019, pp. 459–474.
- [28] Z. Liu, C. Xiong, M. Sun, and Z. Liu, "Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval," in *ACL*, 2018, pp. 2395–2405.
- [29] S. Mai, S. Zheng, and Y. Yang, "Communicative message passing for inductive relation reasoning," in *AAAI*, 2021, pp. 4294–4302.
- [30] T. Mikolov, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013, pp. 1–12.
- [31] T. Rahman, S. Chou, L. Sigal, and G. Carenini, "An improved attention for visual question answering," in *CVPR*, 2021, pp. 1653–1662.
- [32] J. Ren, S. Guo, and F. Chen, "Orientation-preserving rewards' balancing in reinforcement learning," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 6458–6472, 2022.
- [33] A. Sadeghian, M. Armandpour, and P. Ding, "DRUM: end-to-end differentiable rule mining on knowledge graphs," in *NeurIPS*, 2019, pp. 15321–15331.
- [34] H. M. Sergiab, T. Botschen, I. Gurevych, and S. Roth, "A multi-modal translation-based approach for knowledge graph representation learning," in *NAACL*, 2018, pp. 225–234.
- [35] X. Sun, H. Yin, B. Liu, H. Chen, J. Cao, Y. Shao, and N. Hung, "Heterogeneous hypergraph embedding for graph classification," in *WSDM*, 2021, pp. 725–733.
- [36] X. Tang, L. Chen, J. Cui, and B. Wei, "Knowledge representation learning with entity descriptions, hierarchical types, and textual relations," *Inf. Process. Manag.*, pp. 809–822, 2019.
- [37] K. Teru, E. Denis, and W. Hamilton, "Inductive relation prediction by subgraph reasoning," in *ICML*, 2020, pp. 9448–9457.
- [38] G. Wan, S. Pan, and C. Gong, "Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning," in *IJCAI*, 2020, pp. 1926–1932.
- [39] H. Wang, S. Li, and R. Pan, "Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning," in *EMNLP*, 2019, pp. 2623–2631.
- [40] M. Wang, H. Wang, G. Qi, and Q. Zheng, "Richpedia: A large-scale, comprehensive multi-modal knowledge graph," *Big Data Res.*, p. 159–170, 2020.
- [41] P. Wang, J. Han, and C. Li, "Logic attention based neighborhood aggregation for inductive knowledge graph embedding," in *AAAI*, 2019, pp. 7152–7159.
- [42] Z. Wang, L. Li, Q. Li, and D. Zeng, "Multimodal data enhanced representation learning for knowledge graphs," in *IJCNN*, 2019, pp. 1–8.
- [43] Y. Xian and Z. Fu, "Reinforcement knowledge graph reasoning for explainable recommendation," in *SIGIR*, 2019, pp. 285–294.
- [44] R. Xie, S. Heinrich, and M. Sun, "Integrating image-based and knowledge-based representation learning," *IEEE Trans. Cogn. Dev. Syst.*, pp. 169–178, 2020.
- [45] R. Xie, Z. Liu, and M. Sun, "Integrating image-based and knowledge-based representation learning," *IEEE Trans. Cogn. Dev. Syst.*, vol. 12, no. 2, pp. 169–178, 2020.
- [46] W. Xiong, T. Hoang, and W. Wang, "Deeppath: A reinforcement learning method for knowledge graph reasoning," in *EMNLP*, 2017, pp. 564–573.
- [47] X. Xu, P. Zhang, Y. He, and C. Yan, "Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs," in *IJCAI*, 2022, pp. 2341–2347.
- [48] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-supervised learning for recommender systems: A survey," *CoRR*, vol. abs/2203.15876, 2022.
- [49] Q. Zhang, J. Fu, X. Liu, and X. Huang, "Adaptive co-attention network for named entity recognition in tweets," in *AAAI*, 2018, pp. 5674–5681.
- [50] W. Zhang, B. Paudel, and L. Wang, "Iteratively learning embeddings and rules for knowledge graph reasoning," in *WWW*, 2019, pp. 2366–2377.
- [51] Y. Zhang and Q. Yao, "Knowledge graph reasoning with relational digraph," in *WWW*, 2022, pp. 912–924.
- [52] F. Zhao, H. Sun, and L. Jin, "Structure-augmented knowledge graph embedding for sparse data with rule learning," *Comput. Commun.*, pp. 271–278, 2020.
- [53] S. Zheng, W. Wang, and J. Qu, "Mmkgr: Multi-hop multi-modal knowledge graph reasoning," *CoRR*, vol. abs/2209.01416, 2022.
- [54] J. Zhuo, Q. Zhu, and Y. Yue, "A neighborhood-attention fine-grained entity typing for knowledge graph completion," in *WSDM*, 2022, pp. 1525–1533.



Shangfei Zheng received the master's degree in software engineering from Shandong Normal University, China, in 2020. He is currently pursuing the PhD degree with the School of Computer Science & Technology, Soochow University, China. His main research interests include knowledge graph, multi-modal machine learning and reinforcement learning.



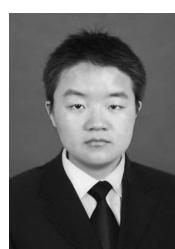
Hongzhi Yin received the PhD degree in computer science from Peking University, in 2014. He works as an ARC Future Fellow, associate professor, and director of the Responsible Big Data Intelligence Lab (RBDI) at The University of Queensland, Australia. He has made notable contributions to predictive analytics, recommendation systems, graph learning, and decentralized and edge intelligence. He has published 250+ papers with an H-index of 62.



Tong Chen received the PhD degree in computer science from the University of Queensland, in 2020. He is currently a lecturer in business analytics with the Data Science Group, School of ITEE, University of Queensland. His research interests include data mining, recommender systems, user behavior modelling, and predictive analytics.



Quoc Viet Hung Nguyen received the Ph.D. degree from EPFL, Switzerland. He is currently a Senior Lecturer with Griffith University, Australia. He has published several articles in top-tier venues, such as SIGMOD, VLDB, SIGIR, KDD, AAAI, ICDE, IJCAI, JVLD, TKDE, TOIS, and TIST. His research interests include data integration, data quality, information retrieval, trust management, recommender systems, machine learning, and big data visualization.



Wei Chen received the PhD degree in computer science from Soochow University, China, in 2018. He is currently an associate professor with the School of Computer Science and Technology, Soochow University, China. His research interests include spatiotemporal data analysis, heterogeneous network analysis, knowledge graph.



Lei Zhao received the PhD degree in computer science from Soochow University, China, in 2006. He is currently a professor with the School of Computer Science and Technology, Soochow University, China. His research interests include graph databases, social media analysis, query outsourcing, parallel, and distributed computing.