Relational Message Passing for Fully Inductive Knowledge Graph Completion

Yuxia Geng  
College of Computer Science and Technology  
Zhejiang University, Hangzhou, China  
gengyx@zju.edu.cn

Jiaoyan Chen  
Department of Computer Science  
The University of Manchester  
Manchester, United Kingdom  
jiaoyan.chen@manchester.ac.uk

Jeff Z. Pan  
School of Informatics  
The University of Edinburgh  
Edinburgh, United Kingdom  
j.z.pan@ed.ac.uk

Mingyang Chen  
College of Computer Science and Technology  
Zhejiang University  
Hangzhou, China  
mimgyangchen@zju.edu.cn

Song Jiang  
NAIE PDU  
Huawei Technologies Co., Ltd.  
Xi'an, China  
jiangsong12@huawei.com

Wen Zhang*, Huajun Chen*  
Zhejiang University, Hangzhou, China  
Donghai Laboratory, Zhoushan, China  
{zhang.wen, huajunsir}@zju.edu.cn

Abstract—In knowledge graph completion (KGC), predicting triples involving emerging entities and/or relations, which are unseen when the KG embeddings are learned, has become a critical challenge. Subgraph reasoning with message passing is a promising and popular solution. Some recent methods have achieved good performance, but they (i) usually can only predict triples involving unseen entities alone, failing to address more realistic fully inductive situations with both unseen entities and unseen relations, and (ii) often conduct message passing over the entities with the relation patterns not fully utilized. In this study, we propose a new method named RMPI which uses a novel Relational Message Passing network for fully Inductive KGC. It passes messages directly between relations to make full use of the relation patterns for subgraph reasoning with new techniques on graph transformation, graph pruning, relation-aware neighborhood attention, addressing empty subgraphs, etc., and can utilize the relation semantics defined in the KG’s ontological schema. Extensive evaluation on multiple benchmarks has shown the effectiveness of RMPI’s techniques and its better performance compared with the existing methods that support fully inductive KGC. RMPI is also comparable to the state-of-the-art partially inductive KGC methods with very promising results achieved. Our codes, data and some supplementary experiment results are available at https://github.com/zjukg/RMPL.

Index Terms—Knowledge Graph, Inductive Knowledge Graph Completion, Message Passing, Link Prediction, Ontology

I. INTRODUCTION

Knowledge Graphs (KGs) [25] often suffer from incompleteness [14]. Many KG completion (KGC) methods, including the schema aware ones [41], have been developed to discover missing facts (triples). Many of them use KG embedding techniques, encoding the KG entities and relations into a vector space with their semantics concerned, so that the missing facts can be inferred by computation on these vector representations (embeddings) [11], [39]. However, these embedding-based methods often work in a transductive setting, where the triples to predict involve only entities and relations that have already occurred in the embedding training triples. When some entities or relations are newly added during testing (a.k.a. unseen entities or relations), they have to re-train the whole KG embeddings, which is not feasible in practice due to the quickly evolving nature and large sizes of many real-world KGs.

Recently, there is an increasing number of inductive KGC studies which aim to complete triples involving unseen entities or unseen relations without training the KG embeddings from the scratch. Among all these works, some try to obtain the embeddings of unseen entities or unseen relations using external resources (e.g., textual descriptions) [15], [26], [29], [36], [42] or auxiliary triples which associate unseen entities with seen entities [5], [17], [38]. Although these approaches can work, the additional resources they require are often unavailable or have low quality, some even with extra high computation costs, e.g. for text embedding.

An promising and widely investigated direction for addressing inductive KGC with unseen entities is to acquire high level semantics merely from the graph structure. The relevant approaches often induce entity-independent logical rules that hold among the relations, from the KG in either statistical [21] or end-to-end differentiable manners [22], [27], [46]. For example, we can induce a rule $\text{father_of}(x, y) \land \text{son_of}(y, z) \rightarrow \text{husband_of}(x, z)$ from the training graph in Fig. 1a, and apply...
it to infer new triples in a testing graph with unseen entities, e.g., $(H, \text{husband_of}, K)$ in Fig. 1b. GraIL [31] is a typical and recent method of this kind, with the scalability and rule expressiveness improved by general graph topological features and a relation-aware graph neural network for message passing and learning over local subgraphs (see more details in Section II-B). GraIL and its follow-up methods [7], [20] have shown the feasibility of using message passing over subgraphs for inductive KGC with promising results achieved on several benchmarks.

However, these methods mostly assume that all the relations in the testing stage are seen with embeddings learned, which is often violated in real-world evolving KGs, especially those constructed via open information extraction systems e.g. NELL [23] and those that are publicly editable e.g. Wikidata [35]. Moreover, they often pass messages directly over entities with the patterns on relations not fully utilized. Although we call the inductive KGC cases with only unseen entities with promising results achieved on several benchmarks.

We name such a fully inductive KGC, to distinguish it from the partially inductive KGC and call those more realistic and more challenging cases with both unseen entities and unseen relations during testing as fully inductive KGC. Fig. 1 provides clear examples on these two cases. Currently, there have been few methods specifically developed for the latter. The only one we know is MaKeR [10] which also relies on the graph structure for prediction.

In this study, we aim to address fully inductive KGC with a method named RMPI, which includes a relation oriented message passing network for subgraph reasoning. It first transforms a triple’s surrounding subgraph in the original KG into a new relation view graph, where inter-relation features are more straightforwardly represented. It then learns the embedding of an unseen relation from the relational subgraph, by the relational message passing network, where novel graph pruning and neighborhood attention techniques are developed for both efficiency and effectiveness, with new neighborhood aggregations being used for addressing the issue of empty subgraphs. As an example, for an unseen relation spouse_of in Fig. 1c, its embedding for predicting the triple with $H$ and $K$ can be obtained by aggregating the embeddings of neighboring relations husband_of, father_of and son_of. Furthermore, RMPI allows the injection of the KG’s ontological schema, which is quite common (see KGs like DBpedia [3] and NELL [23]) and contains complementary relation semantics, for more robust reasoning on fully inductive KGC. In summary, our main contributions are the following:

- We have proposed a robust KGC model named RMPI with novel techniques for effective relational message passing and subgraph reasoning, supporting both partially and fully inductive KGC.
- Extensive experiments have been conducted on 4 newly-constructed benchmarks and 14 public benchmarks from different KGs. RMPI and its variants often outperform the baselines including the state-of-the-art methods.

II. PRELIMINARY

We begin by first formally defining the task and notations, and then briefly introducing the background of subgraph-based inductive reasoning used in existing works.

A. Problem Formulation

A KG is often denoted as $G = \{E, R, T\}$, where $E$ is a set of entities, $R$ is a set of relations, and $T = \{(h, r, t) | h, t \in E, r \in R\}$ is a set of relational facts in form of RDF\(^2\) triple. $h$, $r$ and $t$ are called a triple’s head entity (subject), relation (predicate) and tail entity (object), respectively. KGC is then defined to predict an input candidate triple as true or not (i.e., triple classification), or predict the missing entity/relation in a triple with the other two elements given (i.e., entity/relation prediction), which is often achieved by filling the incomplete triple with a candidate entity/relation and feeding it into the model. It is expected that the true (positive) triples are predicted with higher scores than those false (negative) ones [2]. In a commonly practiced partially inductive KGC setting, a set of unseen entities $E'$ with $E \cap E' = \emptyset$ are newly emerged during prediction, the goal is then to predict a triple $(h, r, t)$ with $r \in R$ and $h, t \in E'$. $G$ is often known as the training graph, while the graph composed of the triples by the unseen entities $E'$ and the relations $R$ is often known as the testing graph.

In this study, we extend the above partially inductive KGC to fully inductive KGC by introducing a set of unseen relations $R'$ with $R \cap R' = \emptyset$ to the testing graph. Specially, we further investigate two situations: one is a general testing graph involving both seen and unseen relations, while the other is a testing graph involving only unseen relations. Formally, a triple given in the testing graph and a triple to predict can be denoted as $(h, r, t)$ with $h, t \in E'$, $r \in (R \cup R')$ or $r \in R'$. We name the evaluation with the first testing graph as testing with semi unseen relations, and name the evaluation with the second testing graph as testing with fully unseen relations.

B. Subgraph-based Partially Inductive Reasoning

To predict a target triple $(h, r, t)$ where $r \in R$, $h, t \in E'$, and the embeddings of $h$ and $t$ are both not available, methods such as GraIL [31] learn and utilize the structural semantics from the subgraph around this triple in an entity-independent manner. To better understand, we re-denote such a triple as $(u, r_t, v)$ using the notations widely used in GNN, where $u$ and $v$ are referred as target head entity and target tail entity, respectively, and $r_t$ is the target relation. The basic workflow of GraIL includes three steps. It first extracts the $K$-hop enclosing subgraph

---

1The term fully inductive has been used in some inductive KGC works that only consider unseen entities [1], [31], meaning the sets of entities seen during training and testing are disjoint, as against the other semi inductive case where unseen entities have to be connected to trained seen entities. In our paper, we name such a fully inductive setting with only unseen entities as partially inductive, to distinguish it from the fully inductive setting we investigated.

2Resource Description Framework. See https://www.w3.org/RDF/.
surrounding the target triple, denoted as $G_{(u, r, v)}$. It then
annotates the entities in the extracted subgraph according to
their shortest distances to $u$ and $v$. Namely, for each entity $i$
in the subgraph, it is labeled with a tuple $(d(i, u), d(i, v))$, where
d$(i, u)$ (resp. $d(i, v)$) denotes the shortest distance between $i$
and $u$ (resp. $v$) without counting any path through $v$ (resp. $u$).
It finally summarizes the subgraph through a GNN encoder and
scores the likelihood of the target triple using the encoded
subgraph representation and the target relation’s embedding.
The GNN-based encoder adopts the general message passing
scheme where a node representation is iteratively updated by
combining it with aggregation of its neighbors’ representations,
and considers the multi-relation features of subgraph. Formally,
the embedding of entity $i$ in the $k$-th GNN layer is given by:

$$ h_i^k = \text{ReLU}( \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \alpha_{ij}^k W_{ij}^r h_j^{k-1} + W_{i,i}^k h_i^{k-1} ) $$

$$ \alpha_{ij}^k = \sigma(A_i^k s + b_i^k) $$

$$ s = \text{ReLU}(A_i^k h_i^{k-1} + b_i^k) $$

where $\mathcal{R}$ is the set of relations in the KG, $\mathcal{N}_i^r$ denotes
the neighbors of entity $i$ under relation $r$, $W_{ij}^r$ is the relation-
specific transformation matrix used in the $k$-th layer, and $W_{i,i}^k$
is the matrix for combining message from itself. $\alpha_{ij}^k$ denotes
the attention weight at the $k$-th layer for the edge connecting $i$
and $j$ via relation $r$, which is computed by a function of the
latent embeddings of $i$ and $j$ learned at previous layer and the
learnable attention embeddings of $r$ and the target relation $r_t$.
The initial embedding $h_i^0$ is represented using the node labels
obtained in the second step (i.e., concatenating the one-hot
vectors of $d(i, u)$ and $d(i, v)$), and the final embedding after
$K$ layers $h_i^K$ is output to generate the subgraph representation
$h_{G_{(u, r, v)}}^K$ and score the triple, as:

$$ \text{score}(u, r_t, v) = W[h_{G_{(u, r, v)}}^K]_u \odot h_{G_{(u, r, v)}}^K \odot h_{G_{(u, r, v)}}^K \odot r_t $$

$$ h_{G_{(u, r, v)}}^K = \frac{1}{|\mathcal{N}_G|} \sum_{i \in \mathcal{N}_G} h_i^K $$

where $\mathcal{N}_G$ is the set of entities in the subgraph. The subgraph is
then represented by averaged pooling over all the output entity
embeddings, and the likelihood of the target triple is scored
through a linear layer with the subgraph representation, the
embedding of the target relation $r_t$, and the output embeddings
of the target entity pair ($h_u^K$ and $h_v^K$), concatenated and fed.
The extracted enclosing subgraph are informative with some
logical evidences (e.g., paths) that could help deduce the
relation between two entities contained. While the distance-
aware entity initial features represent an entity’s relative
positions w.r.t. the target triple’s head and tail entities in
the subgraph, thus capturing the structural semantics without
the need of learning specific embeddings from the whole KG.
Finally, the applied GNN layers pass messages between
entities iteratively to update the entity features, and the resultant
representations are used to score the triple together with the
learnable relation embeddings.

III. METHODOLOGY

A. Overview

In this paper, we make full use of the reasoning clues
reflected by relations, and propose to pass messages (features)
directly from relations to relations iteratively to infer the
likelihood of a target triple, during which the entity features
are not used, while the embedding of an unseen relation can
be inferred and is optionally further augmented by relation
semantics from the KG’s ontological schemas. To achieve this
goal, we proposed a Relational Message Passing Network as
shown in Fig. 2. Briefly, given a target triple, it (i) extracts
an enclosing subgraph $G$ and transforms it to a new graph
that can straightforwardly represent the relationships between
relations via their co-occurrence in $G$, (ii) propagates features
between relations by multiple message passing layers which are
optimized with a graph pruning strategy for higher computation
efficiency and a neighborhood attention mechanism for higher
accuracy, and (iii) computes the target triple’s score, using the
output representation of the target relation, during which the
inductive embeddings of unseen relations are obtained through
the same message passing layers, and they are further allowed to be enhanced by the ontological schema in an apriori manner.

The technical details of the first and the second steps are introduced in the following two subsections — Subgraph Extraction and Transformation and Relational Message Passing, respectively, while the details of the third step are mostly covered by the following Reasoning with Unseen Relations and Triple Scoring and Model Training. Considering the enclosing subgraph is sometimes empty where the relational message passing cannot be applied, we also propose to explore discriminative features from the target triple’s disclosing subgraph for augmentation, which is introduced in the final subsection — Dealing with Empty Subgraphs.

B. Subgraph Extraction and Transformation

Given a target triple $(u, r, v)$, its $K$-hop enclosing subgraph $G$ is extracted by the following steps. First, the $K$-hop incoming and outgoing neighbors of the two target entities $u$ and $v$ from the original KG are collected, denoted as $\mathcal{N}_K(u)$ and $\mathcal{N}_K(v)$, respectively. Then, the enclosing subgraph is generated by taking the intersection of the neighbor sets, i.e., $\mathcal{N}_K(u) \cap \mathcal{N}_K(v)$, pruning the nodes that are isolated or at a distance greater than $K$ from either $u$ or $v$, and adding the triples (i.e., edges) whose head and tail both belong to the pruned entity set. In this way, we obtain a subgraph of distance at most $K + 1$ from the target head to the target tail or vice versa.

$G$ is actually a graph in entity view as the original KG, where the graph nodes correspond to entities and the edges are labeled by relations. To more straightforwardly represent the directed adjacencies between relations, we make a transformation following the idea of line graph [19], that is, a new graph in relation view is generated, during which all edges (identified by the labeled relation and its connected entity pairs) in the original graph are transformed into nodes, and every two new nodes are connected if and only if their corresponding relations are associated with at least one common entity in the original graph. We denote such a relation-view subgraph of $G$ as $R(G)$. A running example is present in Fig. 3 (a) and (b).

Moreover, to represent different connection patterns (relations) of the relations, we review the topological structure of the original entity-view graph and refer to [7] and [10] to define 6 types of edges for the new relation-view graph, as shown in Fig. 3c. For example, the edge $\mathcal{H}=\mathcal{T}$ connecting $r_1$ to $r_2$ is defined since the head entity of relation $r_1$ is also the head entity of relation $r_2$ in the original graph, $\mathcal{T}=\mathcal{H}$ means that the tail entity of $r_1$ is also the head entity of $r_2$; PARA indicates the head entities and the tail entities of $r_1$ and $r_2$ are all identical; while the crossed heads and tails are represented by LOOP. These types of edges model diverse relational connections for the graph in relation view.

C. Relational Message Passing

We next show how to perform message passing on $R(G)$. Since $R(G)$ is much larger and denser than the original graph $G$, conducting message passing over all relation nodes is more computational inefficiency, as theoretically proven by Wang et al. [37]. Motivated by the neighborhood sampling strategy used in GraphSAGE [18] — a GNN model for large scale graphs, which first samples a subset of neighbors for each graph node (up to depth $K$) and then performs message passing on this sampled graph to generate its embedding instead of considering the whole neighborhood. We also consider not updating all the graph nodes in one iteration. Considering that our ultimate goal is to iteratively pass (aggregate) the information for the target relation $r_t$ from its $K$-hop neighbors, we propose to (i) sample nodes on the graph according to whether they fall in the $K$-hop neighborhood of the target relation $r_t$ to generate a tree-like graph with the target relation as the root node, i.e., a target relation-guided graph pruning strategy, and (ii) aggregate and update only the representations that are necessary to satisfy the recursion at each depth.

Algorithm 1 provides detailed pseudocode for illustrating the above procedure. Specifically, given a message passing model
with $K$ GNN layers, we first start from the target relation node and forward sample its full neighborhood sets up to depth $K$ in 1-3 steps. In view of the direction of message passing, we sample all the incoming neighbors for each node. Then, the message passing is conducted layer-by-layer in steps 4-8. In particular, at the $k$-th GNN layer with $k \in \{1, 2, ..., K\}$, we compute the latent representations for only neighbor nodes that are within the $(K - k)$ hops by aggregating the features of their directed neighbors (step 6) and combining the features of themselves to make an update (step 7), as these nodes will contribute their features to the representation learning of the target relation node in the future message passing. For example, when $K = 3$, the first layer computes latent features for nodes that are within 2 hops; while the last layer only aggregates neighborhood features for the root node. Finally, we take the embedding of the target relation output at the last layer, which have already fused the messages from its $K$-hop neighbors, to make a prediction. In this way, we extremely reduce the computation cost with a decreasing number of nodes that participate in the aggregation and are updated at each iteration.

The resulting message passing architecture gives the flexibility to plug in different AGGREGATE and COMBINE functions. In consideration of the diverse connections between relation nodes illustrated by different edge types, we follow the idea of R-GCN [28] to model them in the aggregation. In addition, we also apply a target relation-aware neighborhood attention mechanism to highlight the neighbors that are highly related to the target relation. Formally, the AGGREGATE function at the $k$-th GNN layer is defined as follows:

$$h^k_{N_{r_i}} = \sigma_1 \left( \sum_{e=1}^{6} \sum_{j \in N^e_{r_i}} \alpha^k_{e,r_i,j} W^k_e h^{k-1}_{r_j} \right)$$  \hspace{1cm} (6)

$$\alpha^k_{e,r_i,j} = \frac{\text{exp} (\sigma_2 ((h^{k-1}_{r_j})^T \cdot h^{k-1}_{r_i})))}{\sum_{j \in N^e_{r_i}} \text{exp} (\sigma_2 ((h^{k-1}_{r_j})^T \cdot h^{k-1}_{r_i})))}$$ \hspace{1cm} (7)

where $N_{r_i}$ represents the directed incoming neighbors of node $r_i$, $h^k_{N_{r_i}}$ is the aggregated neighborhood vector; $N^e_{r_i}$ denotes the neighbors under edge type $e$ and $W^k_e$ is the corresponding transformation matrix at the $k$-th layer; $h^{k-1}_{r_j}$ is the representation of a neighbor $r_j$ learned at previous iteration, while $\alpha^k_{e,r_i,j}$ is its attention weight with regard to the target relation $r_i$ in the $k$-th layer, the value is computed by the similarity of the representations of $r_j$ and $r_i$ learned at previous layer, following the assumption that when the neighbor is more related to the target relation, their representations are more similar. A dot-product similarity is adopted here, $\sigma_1$ and $\sigma_2$ are two non linear activation functions, they are ReLU and LeakyReLU (with negative input slope $\alpha = 0.2$), respectively.

After aggregating the neighborhood features, we then combine $r_i$’s representation $h^{k-1}_{r_i}$ obtained at previous layer with the aggregated vector to update its latent representation at the $k$-th layer. The combination is given by:

$$h^k_{r_i} = h^{k-1}_{N_{r_i}} + h^{k-1}_{r_i}$$ \hspace{1cm} (8)

The initial representation of the node at the first layer, i.e., $h^1_{r_i}$, is retrieved from the learnable relation embedding matrix that is randomly initialized. In the last GNN layer, we perform an equal neighborhood aggregation for the target relation $r_i$ to preserve the directed neighborhood information and output its final representation by replacing (6) and (8) as:

$$h^K_{r_i} = \text{ReLU} \left( \sum_{e=1}^{6} \sum_{j \in N^e_{r_i}} W^K_e h^{K-1}_{r_j} \right) + h^{K-1}_{r_i}$$ \hspace{1cm} (9)

D. Reasoning with Unseen Relations

1) Utilizing the graph structure: The above model successfully shows how message passes between relation nodes in the graph, where the features of a given relation node are gained from the representations of its neighboring relations. In testing, for an unseen relation that first appears, we directly generate its embedding by aggregating the embeddings of its neighboring seen relations that have already been learned in training, using the well-trained aggregation functions previously defined, without modifying or re-training the model.

2) Utilizing the ontological schema: A KG is often accompanied by an ontology as its schema for richer semantics and higher quality. For example, the RDF Schema (RDFS)\(^3\)-based ontological schema defines the types of entities (a.k.a. concepts), properties (including object properties i.e. relations, and data properties), concept and relation hierarchies, constraints (e.g., relation domain and range, and concept disjointness). A number of vocabularies built in these ontologies bring in richer semantic relationships (connections) between seen and unseen relations. Therefore, we also investigate utilizing the information from the ontology of a given KG for triple prediction with unseen relations, besides the graph structure.

Given the vocabulary set of RDFS [24], we prefer those that are reasoning relevant and can be used to model correlations between KG relations, and finally select four types of vocabularies with separated relation semantics. They are $\text{rdfs:subPropertyOf}$ for defining the subsumption relationship between relations, $\text{rdfs:domain}$ and $\text{rdfs:range}$ for defining the respective types of head and tail entities of relations, and $\text{rdfs:subClassOf}$ for defining the subsumption relationships between types. All the semantics reflected by them can be represented using RDF triples. For example, $(r_1, \text{rdfs:subPropertyOf}, r_2)$ reveals that $r_1$ is a child relation of $r_2$. These triples constitute a schema graph whose nodes are relations or entity types and edges are vocabularies. A segment can be found in the top left of Fig. 2.

Next, we inject these semantics into our relational message passing network in an apriori manner. Specifically, we first pre-train the schema graph which already covers all the target seen and unseen relations using KG embedding techniques e.g., the method by TransE [6], which can learn vector representations for either seen or unseen relations with their semantic relationships kept in the vector space. Then, we map these relation vectors to participate in the relational message passing to predict the target triple. That is, the projected vectors

\(^3\)https://www.w3.org/TR/rdf-schema/
are taken as the initial representations of nodes in \(R(G)\), i.e., \(\{h^0_{r_i}, \forall r_i \in V\}\). The mapping function is implemented by two fully-connected linear layers, as

\[
h^0_{r_i} = W_1(W_2h^\text{onto})
\]  

(10)

where \(h^\text{onto}\) is the relation vector of \(r\), learned from the schema graph. During testing, the semantic vectors of unseen relations are also projected to make a more robust subgraph reasoning together with those of seen relations.

**E. Triple Scoring and Model Training**

Finally, to obtain the score for the likelihood of the target triple \((u, r, v)\), we pass the output representation of \(r_t\) after \(K\)-layers message passing through a linear layer:

\[
score(u, r, v) = Wh^K_{r_t}
\]  

(11)

Following previous works, the whole model is trained by contrasting the scores of positive and negative triples using e.g. a margin-based ranking loss. In general, each existing triple in the given KGs is taken as a positive triple, while triple samples, have empty enclosing subgraphs, i.e., no valid edge exists in the subgraph under hop \(K\). In this case, it becomes almost impossible to capture graph structure semantics to either infer the relation between a pair of entity or distinguish the positive triples from the negative ones.

Targeting this, we try to explore additional inputs from the \(K\)-hop disclosing subgraph of a target triple. Specifically, we first take the union of the neighbor sets \(N_K(u)\) and \(N_K(v)\) of the target head entity \(u\) and tail entity \(v\), i.e., \(\mathcal{N}_K(u) \cup \mathcal{N}_K(v)\), to generate the \(K\)-hop disclosing subgraph, then convert it to a relation-view subgraph following the same transformation step stated in Section III-B, and finally perform message passing on it to learn supplemental discriminative features for the target triple. In view of the larger size of disclosing subgraph compared with the corresponding enclosing subgraph, we sample the one-hop neighbors of the target relation node and aggregate their features with an attention mechanism which assigns different weights to different neighbors for their importance towards the central target relation node. The aggregation function is given by:

\[
h^d_{N_{r_t}} = \sigma_1(\sum_{r_i \in N'_{r_t}} \alpha^d_{r_i} W^d h^0_{r_i})
\]  

(13)

\[
\alpha^d_{r_i} = \frac{\exp(\sigma_2((W^d h^0_{r_i})^T \cdot (W^d h^0_{r_i})))}{\sum_{r_j \in N'_{r_t}} \exp(\sigma_2((W^d h^0_{r_j})^T \cdot (W^d h^0_{r_j})))}
\]  

(14)

where \(N'_{r_t}\) is the disclosing one-hop neighbor set of the target relation \(r_t\), \(h^0_{rt}\) is the embedding of a neighbor relation \(r_t\), \(W^d\) is a linear transformation matrix applied to every node. Dot-product similarity is adopted to compute the attention value. The activation functions \(\sigma_1\) and \(\sigma_2\) are set to ReLU and LeakyReLU, respectively.

Finally, we integrate the aggregated vector with the representation output from the enclosing subgraph to make more comprehensive triple scoring. Namely, (11) is replaced by either

\[
score(u, r, v) = W(h^K_{r_t} + h^d_{N_{r_t}})
\]  

(15)

with a summation-based fusion function, or

\[
score(u, r, v) = W(h^K_{r_t} \oplus h^d_{N_{r_t}})
\]  

(16)

with a concatenation-based fusion function by the linear transformation matrix \(W_x\).

**IV. EVALUATION**

In our experiments, we first validate that our proposed framework is capable of handling fully inductive KGC utilizing either the graph structure alone or both the graph structure and the ontological schema. Then, we show our proposed relational message passing has superiority in the case of partially inductive KGC with only unseen entities in comparison to existing subgraph reasoning methods. Finally, we perform ablation studies and case studies for more detailed model analyses. We begin by setting up the background for evaluation.

**A. Datasets and Ontologies**

The research on partially inductive KGC with unseen entities is evaluated with a series of benchmarks raised in GraIL [31], which are derived from three famous transductive KGC datasets: WN18RR [12], FB15k-237 [32] and NELL-995 [44]. For robust evaluation, four versions of inductive benchmarks are extracted from each dataset with different sizes. Each benchmark consists of a training graph and a testing graph with disjoint entity sets. In the training graph, 80% of the triples are used to train the model, 10% of them are used as the validation set; while in the testing graph, 10% of the triples are taken as the final targets to predict. The subgraph for each training, validation or testing triple is then extracted from the corresponding training and testing graphs. The statistics are listed in Table Ia.

To support the evaluation of fully inductive KGC, where there exist unseen relations in the testing graph, we hold new benchmarks by re-combining the above 12 inductive benchmarks. Specifically, for an original benchmark, we hold the training graph but replace the testing graph by another benchmark whose testing graph has more relations. For
example, for the second benchmark from NELL-995 (NELL-995.v2 in short), which includes 88 relations, we combine the testing graph of NELL-995.v3 having 122 relations in total, 51 of which are not contained in NELL-995.v2, to generate a new benchmark named as NELL-995.v2.v3. These re-combined datasets are denoted using the pattern “XXX.vi.vj”, where XXX is the source transductive dataset, i is the index indicating which version of inductive benchmark the training graph comes from, while j is the version indicator for the source of the testing graph. Obviously, for each dataset, we filter the testing graph to ensure that all its entities are not present in the corresponding training graph. Since there are no enough unseen relations or testing triples under some combinations, we finally generate 4 new datasets for evaluation, are shown in Table Ib. Notably, the datasets constructed through the above steps actually fall in the setting of testing with semi unseen relations, i.e., a portion of relations in the testing graph have been seen in the training graph while the rest are unseen. Thus, we further extract subsets from the testing graphs by filtering the triples involving seen relations for the setting of testing with fully unseen relations, which are entirely new graphs with only unseen entities and unseen relations. As a result, for each dataset, we obtain two kinds of testing graphs labeled by “TE (semi)” and “TE (fully)”, respectively, as Table Ib shows.

To better address the unseen relations, we also utilize KGs’ ontologies to introduce richer relation semantics. For some KGs that have public ontologies, we can generate relation-aware schema graphs by extracting and re-organizing semantics from these ontologies; while for some KGs that do not have ontologies, it is possible to construct the schema graph from scratch, for example, by inviting experts to set up the relation hierarchy and associate the domain and range constraints of relations. In our work, we experiment with NELL-995 and adopt the schema graph released in [16], which contains 1186 nodes and 3055 triples, covering all the relations and their semantics, and leave the evaluation of the ontology equipped setting on WN18RR and FB15k-237, which have no existing ontologies, as future works. TransE [6] is trained on the schema graph to generate meaningful semantic vectors for both seen and unseen relations.

B. Evaluation and Implementation Details

For more comprehensive evaluation, we tested completion tasks of triple classification and entity prediction. Triple classification predicts an input triple as true or not with a widely used binary classification metric AUC-PR (i.e., area under the precision-recall curve) reported, during which one negative triple is sampled for each triple in the test set that is often positive. While for an incomplete triple (h, r, ?) or (?, r, t), the goal of entity prediction is to rank a set of candidate entities according to their predicted scores of being the tail (or head) entity, and see the rank of the ground truth — the smaller rank, the better performance. Accordingly, the performance is evaluated by widely used metrics of Mean Reciprocal Ranking (MRR) and Hits@n (i.e., the ratio of testing triples whose ground truths are ranked within top-n positions) [39]. For partially inductive KGC, we follow previous works to report Hits@10, while for fully inductive KGC, we make a comprehensive report with Hits@10 and MRR. The negative triples in triple classification and entity prediction are both obtained by replacing the head (or tail) with a random entity. Different from the one-to-one comparisons in triple classification, we follow previous works to rank the ground truth against another 49 randomly sampled candidates in entity prediction. We run each experiment 5 times and report the mean results for a robust comparison.

We implement our model with PyTorch and use Adam as optimizer with learning rate of 0.001, batch size of 16 and margin value of 10, which are the best configurations w.r.t the validation set. For each triple, we sample 2-hop enclosing and disclosing subgraphs, and apply two message passing layers on the enclosing subgraph. For each relation, its embedding size is set to 32, and the dimension of its semantic vector learned from schema graphs is 300. Also, we perform an edge dropout with rate of 0.5. For more details, please refer to our codes.

C. Baselines and Model Variants

1) Baselines: In addition to GraIL as a pioneering method on subgraph-based partially inductive KGC with unseen entities, we also make comparisons with two important state-of-the-art baselines under different settings. The first one is TACT [7], which consists of an entity-based message passing module as GraIL and a relational correlation module which models the topological correlations between the target relations and their adjacent relations and applies a relational correlation network to enrich the representations of the target relations that are omitted by GraIL. These two modules are integrated to compute the score of the target triple. Also, the single relation correlation

| TABLE I: Statistics of the benchmarks. | TR | TE |
|---------------------------------------|----|----|
| WN18RR | #R | #E | #T |
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |

| FB15k-237 | #R | #E | #T |
|----------------|----|----|----|
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |

| NELL-995 | #R | #E | #T |
|---------|----|----|----|
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |

| NELL-995.v2.v3 | #R | #E | #T |
|----------------|----|----|----|
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |

| NELL-995.v3 | #R | #E | #T |
|-------------|----|----|----|
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |

| NELL-995.v2.v3 | #R | #E | #T |
|----------------|----|----|----|
| v1 | 10 | 2746 | 6678 |
| v2 | 10 | 2746 | 6678 |
| v3 | 11 | 2746 | 6678 |
| v4 | 9 | 2746 | 6678 |
TABLE II: Results of fully inductive KGC in the testing with semi unseen relations setting. The best results are marked in bold.

(a) Random Initialized: randomly initializing unseen relations and updating using existing graph structure.

| Methods      | NELL-995.v1.v3 | NELL-995.v2.v3 | NELL-995.v4.v3 | FB15k-237.v1.v4 |
|--------------|----------------|----------------|----------------|-----------------|
|              | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 |
| TACT-base    | 73.98 43.59 72.48 | 85.88 65.63 83.47 | 72.40 52.68 67.95 | 90.29 61.02 82.75 |
| RMPI-base    | 84.06 70.10 82.12 | 89.98 67.98 82.30 | 88.20 70.33 81.20 | 88.76 56.81 79.71 |
| RMPI-NE      | 84.86 56.19 78.63 | 91.10 73.90 88.78 | 83.97 59.47 72.86 | 88.99 57.77 80.38 |

(b) Schema Enhanced: initializing unseen relations using vectors learned from ontological schemas (well support for NELL related datasets).

| Methods      | NELL-995.v1.v3 | NELL-995.v2.v3 | NELL-995.v4.v3 |
|--------------|----------------|----------------|----------------|
|              | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 |
| TACT-base    | 93.20 68.30 89.28 | 93.49 74.41 90.26 | 93.53 74.89 90.25 |
| RMPI-base    | 92.96 70.72 90.34 | 94.31 80.37 92.05 | 94.16 79.20 89.60 |
| RMPI-NE      | 92.50 72.24 91.08 | 94.99 80.03 92.94 | 93.44 72.69 88.53 |

Following the original evaluation in MaKeR, the testing triples are divided into three sets: (i) \( u_{ent} \) where all the entities are unseen while all the relations are seen; (ii) \( u_{rel} \) where all the entities are seen while all the relations are unseen; and (iii) \( u_{both} \) where each triple to predict involves an unseen relation and at least one unseen entity. We follow [10] to report results on entity prediction with metrics of MRR and Hits@10.

2) RMPI Variants: To evaluate the effectiveness of two techniques — the one-hop neighborhood aggregation in the disclosing subgraph for handling empty enclosing subgraphs (NE), and the target relation-aware neighborhood attention in message aggregation (TA), we tested several RMPI variants: RMPI-base without NE and TA, RMPI-NE with NE, RMPI-TA with TA, and RMPI-NE-TA with both NE and TA.

D. Main Results of Fully Inductive KGC

We have two settings to initially represent an unseen relation in the testing graph: (i) random initialization, and (ii) projection from the relation’s semantic vector learned from the ontological schema. These two settings are denoted as Random Initialized and Schema Enhanced, respectively. Both of them are tested with TACT-base, RMPI-base and RMPI-NE, while RMPI-TA with TA, and RMPI-NE-TA is not applicable here.

1) Test with semi unseen relations: The results of fully inductive KGC under the test setting of semi unseen relations are shown in Table II. From Table IIa, we can observe that our models always outperform TACT-base under the setting of Random Initialized. This shows the effectiveness of our proposed relational message passing network which passes messages between relations with multiple aggregation layers, through which the inductive embeddings of unseen relations can be well learned by aggregating their neighboring seen relations. In contrast, the relation correlation module in TACT-base actually only performs one-layer neighborhood aggregation for the central target relation, while for unseen relations that exist in the neighborhood, it cannot provide meaningful representations. The only exception raised in FB15k-237.v1.v4 may be because that (i) the proportion of unseen relations in its testing graph is relatively low (174 seen relations vs. 26 unseen relations), where the negative impact of randomly initialized unseen relations is relatively limited; and (ii) the one-hop neighboring relations are informative enough for predicting the triple.
TABLE III: Results of fully inductive KGC in the testing with fully unseen relations setting. The best results are marked in bold. (a) Random Initialized: randomly initializing unseen relations and updating using existing graph structure.

| Methods     | NELL-995.v1.v3 | NELL-995.v2.v3 | NELL-995.v3.v4 | FB15k-237.v1.v4 |
|-------------|----------------|----------------|----------------|-----------------|
|             | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 |
| TACT-base   | 73.47 34.84 52.64 | 77.78 40.85 68.75 | 67.46 41.80 59.23 | 58.77 13.74 15.57 |
| RMPI-base   | 83.14 52.78 79.09 | 86.18 52.48 70.00 | 82.09 55.14 65.21 | 57.74 13.74 15.57 |
| RMPI-NE     | 84.40 55.39 82.57 | 87.93 61.01 81.51 | 79.38 50.62 67.10 | 60.23 21.36 25.82 |

(b) Schema Enhanced: initializing unseen relations using vectors learned from ontological schemas (well support for NELL related datasets).

| Methods     | NELL-995.v1.v3 | NELL-995.v2.v3 | NELL-995.v3.v4 |
|-------------|----------------|----------------|----------------|
|             | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 | AUC-PR MRR Hits@10 |
| TACT-base   | 92.66 68.55 89.09 | 91.87 62.19 86.70 | 90.80 62.54 80.72 |
| RMPI-base   | 92.56 70.00 90.23 | 93.29 73.36 90.20 | 91.26 69.47 81.33 |
| RMPI-NE     | 92.95 71.82 91.18 | 94.19 74.80 92.09 | 90.09 64.25 81.44 |

TABLE IV: Comparison results with MaKEr [10] on its developed benchmarks (unseen relations are randomly initialized in our models).

| Methods     | u_ent | u_rel | u_both | FB-Ext | u_ent | u_rel | u_both | NELL-Ext |
|-------------|-------|-------|--------|--------|-------|-------|--------|----------|
|             | MRR   | Hits@10 | MRR   | Hits@10 | MRR   | Hits@10 | MRR   | Hits@10 |
| MaKEr       | 74.64 | 95.28 | 32.00 | 50.00 | 77.09 | 94.64 | 31.53 | 60.00 |
| RMPI-base   | 50.48 | 82.55 | 51.61 | 69.50 | 76.36 | 94.85 | 46.09 | 82.50 |
| RMPI-NE     | 53.51 | 82.85 | 51.47 | 69.00 | 75.46 | 96.05 | 63.99 | 89.17 |

TABLE V: Comparison results with MaKEr [10] on NELL-Ext when our models are enhanced by ontological schema.

| Methods     | u_ent | u_rel | u_both |
|-------------|-------|-------|--------|
|             | MRR   | Hits@10 | MRR   | Hits@10 |
| MaKEr       | 77.09 | 94.64 | 31.53 | 60.00 |
| RMPI-base   | 76.52 | 94.92 | 66.23 | 95.83 |
| RMPI-NE     | 75.90 | 95.19 | 75.16 | 99.17 |

When the representations of the unseen relations are augmented by the ontological schema, the performance of both our models and TACT-base is greatly improved, as shown in Table IIb. This validates our insight of enriching the neighborhood connections between seen and unseen relations by ontological schemas. Meanwhile, RMPI-base and RMPI-NE both achieve better or comparable performance against TACT-base, which is consistent with the results under Random Initialized.

2) Test with fully unseen relations: The results of fully inductive KGC under the test setting of fully unseen relations are shown in Table III. It can be seen that the performance dramatically drops when performing random initialization as there are no seen relations around to provide meaningful information for updating the representations of unseen relations. The only clue for reasoning may lie in the high-level graph patterns, e.g., the co-occurrence patterns of relations, while our proposed models are better at capturing such patterns than TACT-base, which can be verified by the less performance degradation, especially on NELL-995.v1.v3 and NELL-995.v2.v3.

3) Comparison with MaKEr: The results of Random Initialized are reported in Table IV while the results of Schema Enhanced are reported in Table V. On the testing sets of u_rel and u_both, our models always achieve better results than MaKEr, even though the unseen relations are randomly initialized, and the performance is further improved when augmented by the ontological schema. While on the testing set of u_ent, our models gain slightly superior Hits@10 and comparable MRR on NELL-Ext, but gain largely inferior MRR and Hits@10 on FB-Ext. This may be attributed to (i) the high coverage of seen relations in MaKEr’s testing graphs, that is, there are 6103 (resp. 2100) triples in the testing graph of FB-Ext (resp. NELL-Ext) used for providing the local graph structures, 5713 (resp. 1923) out of which contain seen relations, thus the testing on u_ent is actually a setting close to the partially inductive KGC; and (ii) the nature of relations in FB15k-237, i.e., the directed neighbors are informative enough to predict the triple. Overall, our proposed message network accomplishes promising results in the fully inductive case compared with the mete learning based MaKEr.

E. Main Results of Partially Inductive KGC

Our method RMPI is specifically developed for fully inductive KGC, but is also compatible to partially inductive KGC. Thus we also evaluate the variants of RMPI for partially inductive KGC by comparing them with the state-of-the-art subgraph reasoning-based methods including GraIL, TACT, TACT-base and CoMPILE. The results of entity prediction are shown in Table VIa. We can see that our models always outperform the baselines. Especially, our models achieve a large margin of improvements on the series of datasets from WN18RR and NELL-995.v4. The results of triple classification are shown in Table VIb. Although our models are not dominated, they are still the second best or comparable to the second best. These results indicate the superiority of our relational message passing network against these baselines which mainly rely on the entities for message passing. Moreover, regarding the less space for improvement in triple classification task (i.e., the AUC-PR values are more than 90% and close to 100% in most situations), we prefer to claim our superiority on the more challenging entity prediction task.

F. Ablation Studies

1) Different Components: We investigate the contributions of the one-hop neighborhood aggregation in disclosing subgraphs (NE) and the target relation-aware neighborhood attention in
message passing (TA) by comparing the variants of RMPI. First, for NE, we find that RMPI-NE improves RMPI-base in most situations in the fully inductive KGC cases, especially in testing with fully unseen relations with Random Initialization, illustrating that the additional inputs from the disclosing graph provide more graph patterns for prediction. The improvements vary from case to case, this could be attributed to the datasets and their triples for prediction. The information from the enclosing subgraph sometimes is enough to make a reasonable prediction. In the partially inductive case, RMPI-NE and RMPI-NE-TA outperform RMPI-base and RMPI-TA, respectively, on either entity prediction or triple classification in most situations, as shown in Table VIa and Table VIb. In particular, for the entity prediction on WN18RR derived benchmarks, where many negative triples and quite a few positive triples have empty enclosing subgraphs, NE plays a significant positive role, e.g., RMPI-NE-TA (RMPI-NE resp.) achieves an averaged 7.85% (7.40% resp.) improvement over RMPI-TA (RMPI-base resp.). To sum up, with the technique of NE, the disclosing graph is able to provide discriminative features when the enclosing subgraph are absent, and complementary semantics when the enclosing subgraph exists.

Second, we verify the effectiveness of TA by comparing RMPI-base against RMPI-TA, and RMPI-NE against RMPI-NE-TA. From Table VIa and Table VIb, we can observe that the variants accompanied by TA perform better in most situations. For example, on the benchmarks derived from FB15k-237, the average improvement of RMPI-TA over RMPI-base on entity prediction is 1.41%, and that of RMPI-NE-TA over RMPI-NE is 1.81%. On triple classification, the corresponding improvements are 0.84% and 0.89%, respectively. These results suggest that it is necessary to consider the importance of different neighboring relations w.r.t. the target relation, especially for those relations that are more than one hop away. However, this mechanism may not work well for some datasets such as NELL-995.v4. It may be because some distant relations that are important for reasoning are filtered out due to their less relatedness to the target relation. In the future, we will study more robust mechanisms for TA.

2) Different Fusion Functions: To integrate the information from the enclosing and disclosing subgraphs, we design the summation-based fusion function (15) and the concatenation-based fusion function (16). Both functions can advance our model for better performance (see the analysis on RMPI-NE). Table VII presents the results of these two functions in different cases for comparison. We can find that the better function varies from dataset to dataset, from setting to setting and from task to task. In the future, we will investigate more robust fusion functions that can generalise to different situations.

3) Ontological Schemas: In fully inductive KGC, the extra relation semantics from ontological schemas consistently improve the performance of RMPI as we have presented. While in partially inductive KGC, we also investigate the impact of using ontological schemas. The results of our models and the baseline TACT-base, with and without schemas, on two benchmarks derived from NELL-995 are shown in Table VIII. We can observe that the performance is augmented in most situations, especially for TACT-base on NELL-995.v4. This large improvement on NELL-995.v4 may be due to the compact relatedness of the relations. TACT-base only models the part of relation correlations, while the ontological schema provides relatively important complementary relation semantics for it. All of these results illustrate the great significance of utilizing external relation semantics from the ontological schema.

G. Case Studies

In Fig. 4, we present two positive target triples to be predicted, as well as their enclosing subgraphs, the relations in their neighborhoods, and their scores predicted by different models. The first example includes an unseen relation coach
As we can see, there are both seen and unseen relations in the neighborhood, while our models are more robust to perform reasoning. More specifically, the one-hop neighboring relation \textit{team won trophy} that could be used to infer the target relation \textit{works for} is also an unseen relation whose embedding is not available, while the 2-hop seen relations such as \textit{team also known as} and \textit{team against team} that can be utilized by our models could imply that \textit{team won trophy} is a \textit{team} related relation and may further infer that a team and its one member (implied by \textit{works for}) tend to own the same thing. When referring to the ontological schema where the seen and unseen relations are connected to some shared concepts via e.g. domain and range relationships, the real meanings of the unseen relations can be figured out, and more spaces are exposed for reasoning, thus resulting in improved prediction scores of TACT-base and RMPI-base. From the second example, we can see that the one-hop neighboring relations /music/genre/parent_genre and /music/genre/artists are informative enough to make a decision, that is, an artist falling in the child music genre also is labeled by the parent genre, while the 2-hop neighbors are noisy to collaborate. As a result, TACT-base performs better on predicting it than RMPI-base. However, the target relation-aware attention mechanism applied in the RMPI-TA improves the score of the second triple by highlighting the relations in the remote neighborhood and filtering those that are noisy for prediction.

## V. Related Work

### A. Transductive Knowledge Graph Completion

Existing transductive KG completion methods usually assign a unique vector to each entity and relation and train it using the observed triple facts. During evaluation, the embeddings of entities and relations in a testing triple are then looked up to score its plausibility. According to how a triple is scored, these methods can be grouped into three categories: translation-based ones such as TransE [6], TransH [40] and RotatE [30], \textit{semantic matching}-based ones such as DistMult [45] and ComplEx [33], and \textit{neural network}-based ones such as ConvE [13]. Recently, inspired by the great success of graph neural networks (GNNs) in processing graph structured data, some works employ relation-aware GNNs as powerful encoders to encode the structural information contained in KGs. Typical practices include RGCN [28], CompGCN [34], etc.

All of these models have shown their capability to capture complex semantic patterns in KGs and achieve state-of-the-art completion results in the transductive setting. However, they fail to complete triples involving unseen entities and/or relations that have never appeared in the triples used for training.

### B. Inductive Knowledge Graph Completion

1) Partially Inductive: The partial inductive KG settings address either the unseen entities or unseen relations. Several lines of inductive KG methods have been proposed to learn embeddings for unseen entities following different settings [9].

One line of them use entity’s external resources such as text and images as additional inputs [8], [29], [42], [43]. For example, to incorporate with the textual information, early works first learn a continuous text embedding for each entity by e.g. averaging pre-trained word embeddings [29] or applying deep convolutional neural models [42] for tokens in its name or description, and then map the text embeddings to compute the triple score together with the geometric embeddings learned through traditional transductive KG methods. The embeddings of unseen entities can thus be obtained via applying the learned mapping. Instead of learning two kinds of representations, some recent works directly use the text resources to predict triples. For example, KG-BERT [47] and StAR [36], which build upon pre-trained language models such as BERT, model the KG translation-based ones such as TransE [6], TransH [40] and ComplEx [33], and \textit{semantic matching}-based ones such as DistMult [45] and ComplEx [33], and \textit{neural network}-based ones such as ConvE [13].

Another line of works consider a few auxiliary triples of an emerging entity. These triples link emerging entities to seen entities that have known to the KG and serve as the context to compute their embeddings. For example, Hamaguchi et al. [17] apply a powerful GNN to generate the embedding of a new entity by aggregating all its neighboring known entities. The subsequent works [4], [5], [38] propose different neighborhood aggregation functions for optimizing this process.

Different from the above works which rely on entity’s external resources or auxiliary triples, some efforts have been made to process the unseen entities solely based on the existing
graph structure by inducing probabilistic logical rules from the KG, which capture entity-independent relational semantics and are inherently inductive [21], [22], [27], [46]. In particular, Teru et al. [31] propose GraIL, which aims to implicitly induce the logical rules by reasoning over the the enclosing subgraph surrounding a candidate triple in an entity-independent manner. The follow-up works improve this process by introducing the topological correlations between relations [7] or optimizing the message passing in a node-edge interaction manner [20]. While in this paper, we focus on the information of relations and conduct relational message passing to predict triples.

However, the methods introduced above mainly aim at unseen entities and require all the testing relations have been seen at training stage. There are also some works aiming at unseen relations. For example, ZSGAN [26] and OntoZSL [15] leverage generative adversarial networks (GANs) to synthesize valid inductive embeddings for unseen relations conditioned on their external resources such as textual descriptions and ontological schemas. However, they are also partially inductive with the constraints that each entity that appears in the testing set has appeared during training (i.e., all the entities are seen).

1) Partially Inductive: The partially inductive KGC settings aim to address unseen entities and unseen relations simultaneously. Such settings are quite common in involving KGs but are more challenging and relatively underexplored. One of them is a simple extension of the text-based partially inductive method, which also provides text resources for unseen relations as for unseen entities [36]. However, the extra computation cost on massive text resources, especially using large-scale pre-trained language models, is actually a serious turn-off.

Another work that is more related to ours is MaKEr [10], a method that also depends on the local graph structure to deal with the unseen entities and unseen relations. In comparison to the meta learning based framework used in MaKEr, in our paper, we instead perform reasoning over the enclosing subgraph of a candidate triple using a message passing network. Moreover, MaKEr actually requires that a part of the relations in the local neighborhood to be seen, and only supports the setting of testing with semi unseen relations. In contrast, our method can support this setting as well as another more challenging setting of testing with fully unseen relations where all the relations in the local subgraph are unseen.

### VI. Conclusion and Outlook

In this paper, we propose RMPI, a relational message passing network for subgraph-based inductive reasoning, which focuses on the relations in the local subgraph and directly passes messages from relations to relations iteratively to infer missing triples in a KG. It supports not only partially inductive KGC with only unseen entities but also the more challenging problem of fully inductive KGC. To implement RMPI, we (i) developed an effective graph pruning strategy to improve the reasoning efficiency; (ii) applied a target relation-aware neighborhood attention mechanism to better aggregate features from the neighborhood with edge types of the relational graph utilized; (iii) utilized discriminative features from the surrounding disclosing subgraphs to deal with empty enclosing subgraphs; and (iv) injected the KG’s ontological schema for richer relation semantics. In evaluation, we have performed extensive experiments on many different benchmarks with baselines of state-of-the-art or typical partially inductive and fully inductive KGC methods including TACT, MaKEr, GraIL, CoMPLEx and so on. RMPI often achieves better performance than the baselines, and the effectiveness of its technical components, such as the injection of ontological schema, the target relation-aware neighborhood attention mechanism, and the aggregation on disclosing subgraphs, has been fully verified.

In the future, we plan to study the following aspects: 1) developing more robust solutions to integrate the ontological schema for inductive KGC; 2) assembling nonnegligible reasoning clues from entities for relational message passing; and 3) performing more comprehensive evaluations on fully inductive KGC with both seen and unseen entities.

### Acknowledgment

This work is partially funded by NSFCU19B2027/91846204, Zhejiang Provincial Natural Science Foundation of China (No. Q23F020051), the EPSRC project ConCur (EP/V050869/1) and the Chang Jiang Scholars Program (J2019032).
REFERENCES

[1] Mehdi Ali, Max Berrendorf, Mikhail Galkin, Veronika Thost, Tengfei Ma, Volker Tresp, and Jens Lehmann. Improving inductive link prediction using hyper-relational facts. In ISWC, pages 74–92, 2021.

[2] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Negative statements considered useful. joural of Web Semantics, 71, 2021.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In The semantic web, pages 722–735. Springer, 2007.

[4] Jinhoon Baek, Dong Bok Lee, and Sung Ju Hwang. Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. Advances in Neural Information Processing Systems, 33:546–560, 2020.

[5] Rajarshi Bhoskam and Gerard de Melo. Explainable link prediction for emerging entities in knowledge graphs. In ISWC (1), volume 12506 of Lecture Notes in Computer Science, pages 39–55. Springer, 2020.

[6] Antoine Bord, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakshenko. Translating embeddings for modeling multi-relational data. Advances in Neural Information Processing Systems, 26, 2013.

[7] Jiajun Chen, Huauru He, Feng Wu, and Jie Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 6271–6278, 2021.

[8] Jiaoyan Chen, Yuxia Geng, Zhao Chen, Yuan Horrocks, Jeff Z. Pan, and Huajun Chen. Knowledge-aware zero-shot learning: Survey and perspective. In IJCAI Survey Track, 2021.

[9] Jiaoyan Chen, Yuxia Geng, Zhao Chen, Jeff Z. Pan, Yuan He, Wen Zhang, Ian Horrocks, and Huajun Chen. Low-resource learning with knowledge graphs: A comprehensive survey. unpublished.

[10] Tingyang Chen, Wen Zhang, Zhen Yao, Xuanhan Chen, Mengxiao Ding, Fei Huang, and Huajun Chen. Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In IJCAI, 2022.

[11] Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongqiao Duan. Knowledge graph completion: A review. IEEE Access, 8:192435–192456, 2020.

[12] Tim Dettmers, Pasquale Minervini, Sebastian Riedel, and Christian Meilicke. Dynamic and convolutional knowledge graph embeddings. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[14] Jiaoyan Chen, Pasquale Minervini, Thiago M. Neves, and Christian Meilicke. Knowledge-aware zero-shot learning: Survey and perspective. In IJCAI Survey Track, 2021.

[15] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Toml. Knowledge graph embedding by translating on hyperplanes. In AAAI, pages 1715–1721, 2019.

[16] Lukas Hutter, hears, Volker Tresp, and Jens Lehmann. Improving inductive link prediction using hyper-relational facts. In ISWC, pages 74–92, 2021.

[17] Christian Meilicke, Manuel Fink, Yanzhe Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In International Semantic Web Conference, pages 3–20. Springer, 2018.

[18] Tim Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. Communications of the ACM, 61(5):103–115, 2018.

[19] Jeff Z. Pan. Resource description framework. In Handbook on Ontologies, pages 71–90. Springer, 2009.

[20] Jeff Z. Pan, Guido Vetere, Jose Manue Gomez-Perez, and Honghan Wu. Exploiting Linked Data and Knowledge Graphs in Large Organisations. Springer, 2017.

[21] Pengda Qin, Xin Wang, Wenhui Chen, Chunyang Zhang, Weiran Xu, and William Yang Wang. Generative adversarial zero-shot relational learning for knowledge graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8673–8680, 2020.

[22] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. Advances in Neural Information Processing Systems, 32, 2019.

[23] Michael Sejr Schlüchtermann, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In ESWC, pages 593–607, 2018.

[24] Haseeb Shah, Johannes Villmow, Adrian Uges, Ulrich Schwanecke, and Faisal Shaftai. An open-world extension to knowledge graph completion models. In AAAI, pages 3044–3051, 2019.

[25] Zhaqing Sun, Zhi-Hong Deng, Juan-Yun Nie, and Jian Tang. RotAE: Knowledge graph embedding by relational rotation in complex space. In ICLR (Poster), OpenReview.net, 2019.

[26] Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by inductive reasoning. In International Conference on Machine Learning, pages 9448–9457. PMLR, 2020.

[27] Kristina Toutanova, Danqi Chen, Patrick Pantel, Houfeng Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In EMNLP, pages 1499–1509, 2015.

[28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In International Conference on Machine Learning, pages 2071–2080. PMLR, 2016.

[29] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. Composition-based multi-relational graph convolutional networks. In ICLR. OpenReview.net, 2020.

[30] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10):78–85, 2014.

[31] PeiFei Yang, Zhi Hong Han, ChenLiang Li, and Rong Pan. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7152–7159, 2019.

[32] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12):2724–2743, 2017.

[33] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of knowledge graph embeddings. In IJCAI, pages 1802–1808, 2017.

[34] Quan Wang, Zhe Deng, Fei Wang, and Lin Gao. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12):2724–2743, 2017.

[35] Ioannis Kornilakis, Konstantinos Tzortzis, and John S. Vitter. Knowledge graph completion. In IJCAI, pages 1802–1808, 2017.

[36] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Recursive deep models for semantic role labeling. In EMNLP, pages 564–573, 2017.

[37] Pengda Qin, Xin Wang, Wenhui Chen, Chunyang Zhang, Weiran Xu, and William Yang Wang. Zero-shot learning. In Journal of Web Semantics, volume 33, pages 7152–7159, 2019.

[38] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. A reinforcement learning method for knowledge graph reasoning. In EMNLP, pages 2071–2080, 2017.

[39] Pengda Qin, Xin Wang, Wenhui Chen, Chunyang Zhang, Weiran Xu, and William Yang Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. Advances in Neural Information Processing Systems, 32, 2019.

[40] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. Composition-based multi-relational graph convolutional networks. In ICLR. OpenReview.net, 2020.

[41] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10):78–85, 2014.

[42] Pengda Qin, Xin Wang, Wenhui Chen, Chunyang Zhang, Weiran Xu, and William Yang Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. Advances in Neural Information Processing Systems, 32, 2019.

[43] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. Advances in Neural Information Processing Systems, 30, 2017.

[44] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for knowledge graph completion. unpublished.