Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs

Jiajun Chen, Huarui He, Feng Wu, Jie Wang*
University of Science and Technology of China
{jiajun98, huaruihe}@mail.ustc.edu.cn, {fengwu, jiewangx}@ustc.edu.cn

Abstract

Inductive link prediction—where entities during training and inference stages can be different—has been shown to be promising for completing continuously evolving knowledge graphs. Existing models of inductive reasoning mainly focus on predicting missing links by learning logical rules. However, many existing approaches do not take into account *semantic correlations* between relations, which are commonly seen in real-world knowledge graphs. To address this challenge, we propose a novel inductive reasoning approach, namely TACT, which can effectively exploit **T**opology-**A**ware **C**orrela**T**ions between relations in an entity-independent manner. TACT is inspired by the observation that the semantic correlation between two relations is highly correlated to their topological structure in knowledge graphs. Specifically, we categorize all relation pairs into several *topological patterns*, and then propose a Relational Correlation Network (RCN) to learn the importance of the different patterns for inductive link prediction. Experiments demonstrate that TACT can effectively model semantic correlations between relations, and significantly outperforms existing state-of-the-art methods on benchmark datasets for the inductive link prediction task.

Introduction

Knowledge graphs store quantities of structured human knowledge in the form of factual triples, which have been widely used in many fields, such as natural language processing (Zhang et al. 2019), recommendation systems (Wang et al. 2018), and question answering (Huang et al. 2019).

For real-world knowledge graphs, new entities keep emerging continuously, such as new users and products in e-commerce knowledge graphs and new molecules in biomedical knowledge graphs (Teru, Denis, and Hamilton 2020). Moreover, they usually face the incompleteness problem, i.e., some links are missing. To address this challenge, researchers have paid increasing attention to the inductive link prediction task (Tran et al. 2016; Hamaguchi et al. 2017; Wang et al. 2019; Teru, Denis, and Hamilton 2020). Inductive link prediction aims at predicting missing links between entities in knowledge graphs, where entities during training and inference stages can be different. Despite the importance of inductive link prediction in real-world applications, many existing works focus on transductive link prediction and cannot manage previously unseen entities (Sadeghian et al. 2019). Inductive link prediction is challenging as we need to determine which relation it is between two unseen entities during training, that is, we need to generalize what is learned from training entities to unseen entities.

Existing models of inductive reasoning mainly focus on predicting missing links by learning logical rules in knowledge graphs. Rule learning based methods (Yang, Yang, and Cohen 2017; Sadeghian et al. 2019) explicitly mine logical rules based on observed co-occurrence patterns of relations, which are inherently inductive as the learned rules are entity-independent and can naturally generalize to new entities. More recently, GraIL (Teru, Denis, and Hamilton 2020) implicitly learns logical rules with reasoning over subgraph structures in an entity-independent manner. However, many existing inductive reasoning approaches do not take into account neighboring relational triples when predicting the missing links.

To take advantage of the neighboring relational triples, we exploit *semantic correlations* between relations, which are commonly seen in knowledge graphs. For example, for the relations in Freebase (Bollacker et al. 2008), "/people/person/nationality" and "/people/ethnicity/languages_spoken" have a strong correlation as a person's spoken languages are correlated to the person's nationality, while "/people/person/nationality" and "/film/film/country" have a weak correlation. Moreover, the topological patterns between any two relations—the way how they are connected in knowledge graphs—can be different, which influence the correlation patterns. For example, for the relation pair "father_of" and "has_gender" in Figure 1, they are connected by e_1 in a tail-to-tail manner and connected by e_2 in a head-to-tail manner, which are different topological patterns.

In this paper, we propose a novel inductive reasoning approach, namely TACT, which can effectively exploit **T**opology-**A**ware **C**orrela**T**ions between relations in knowledge graphs. Specifically, TACT models the semantic correlations in two aspects: correlation patterns and correlation coefficients. We categorize all relation pairs into several

*Corresponding author.

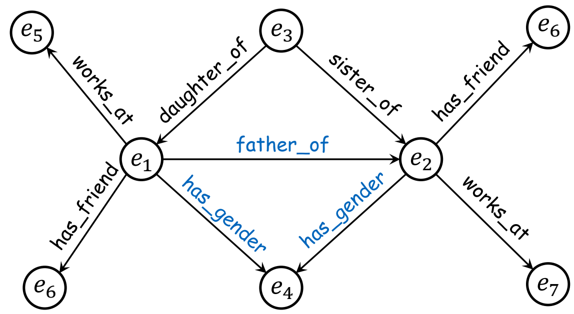

Figure 1: An example in knowledge graphs.

different correlation patterns according to their topological structures. We then convert the original knowledge graph to a Relational Correlation Graph (RCG), where the nodes represent the relations and the edges indicate the correlation patterns between any two relations in the original knowledge graph. Based on the RCG, we propose a Relational Correlation Network (RCN) to learn the correlation coefficients of the different patterns for inductive link prediction. TACT can effectively incorporate the information of neighboring relations, thus to promote the performance of link prediction in the inductive setting. Experiments demonstrate that TACT can effectively model semantic correlations between relations, as well as significantly outperforms existing state-of-the-art methods on benchmark datasets for the inductive link prediction task.

Notations Given a set $\mathcal{E}$ of entities and a set $\mathcal{R}$ of relations, a knowledge graph $\mathcal{G} = \{(u, r, v) | u, v \in \mathcal{E}, r \in \mathcal{R}\}$ is a collection of facts, where u, v, and r represent head entities, tail entities, and relations between head and tail entities, respectively. We use $\mathbf{e}_u, \mathbf{r}, \mathbf{e}_v$ to denote the embedding of the head entity, the relation and the tail entity. Let d denote the embedding dimension. We denote i-th entry of a vector $\mathbf{e}$ as $[\mathbf{e}]_i$. Let $\circ$ be the Hadamard product between two vectors,

$$[\mathbf{a} \circ \mathbf{b}]_i = [\mathbf{a}]_i \cdot [\mathbf{b}]_i,$$

and we use $\oplus$ to denote the concatenation of vectors.

Related Work

Rule Learning Based Methods Rule learning based methods learn logical rules based on observed co-occurrence patterns of relations, which is inherently inductive as the learned rules are independent of entities. Mining rules from data is the central task of inductive logic programming (Muggleton 1992). Traditional methods suffer from the problem of scaling to large datasets or being challenging to optimize. More recently, Neural LP (Yang, Yang, and Cohen 2017) proposes an end-to-end differentiable framework to learn both the structure and parameters of logical rules. DRUM (Sadeghian et al. 2019) further improves Neural LP with mining more correct logical rules. However, rule learning based methods mainly focus on mining horn rules, which limits their ability to model more complex semantic correlations between relations in knowledge graphs.

Embedding Based Methods Knowledge graph embedding has been shown to be a promising direction for knowledge graph reasoning (Sun et al. 2019; Zhang et al. 2020a; Zhang, Cai, and Wang 2020). Some embedding based methods can generate embeddings for unseen entities. Hamaguchi et al. (2017) and Wang et al. (2019) learn to generate entity embeddings for unseen entities by aggregating neighbor entity embeddings with graph neural networks. However, they need new entities to be surrounded by known entities, which cannot handle entirely new graphs. GraIL (Teru, Denis, and Hamilton 2020) develops a graph neural network based link prediction framework that reasons over local subgraph structures, which can perform inductive link prediction in an entity-independent manner. However, GraIL fails to model semantic correlations between relations, which are common in knowledge graphs.

Link Prediction with GNNs In recent years, graph neural network (Kipf and Welling 2017; Velickovic et al. 2018) shows great potential in link prediction, as knowledge graphs naturally have graph structures. Schlichtkrull et al. (2018) propose a relational graph neural network to consider the connected relations when applying aggregation on the entities. More recently, Zhang et al. (2020b) propose a relational graph neural network with hierarchical attention to effectively utilize the neighborhood information of entities in knowledge graphs. However, those methods have difficulty in predicting missing links between unseen entities, as they do link prediction relying on the learned entity embeddings during training.

Modeling Correlations Between Relations Several existing knowledge graph embedding methods consider the problem of modeling correlations between relations. Do, Tran, and Venkatesh (2018) decompose the relation–specific projection spaces into a small number of spanning bases, which are shared by all relations. Zhu et al. (2018) propose to learn the embedded relation matrix by decomposing it as a product of two low-dimensional matrices. Different from the aforementioned work, our work

(a) innovatively categorizes all relation pairs into seven *topological patterns* and propose a novel relational correlation network to model *topology-aware* correlations.

(b) considers the inductive link prediction task, while the aforementioned knowledge graph embedding methods have difficulty in dealing with it.

(c) outperforms existing state-of-the-art inductive reasoning approaches on benchmark datasets.

Methods

In this section, we introduce our proposed method TACT. To perform inductive link prediction, TACT aims at scoring a given triple (u, r_t, v) in an entity-independent way, where r_t is the target relation between the entity u and v. Specifically, TACT consists of two modules: the relational correlation module and the graph structure module. The relational correlation module is proposed based on the observation that semantic correlations between any two relations are highly correlated to their topological structures, which are commonly seen in knowledge graphs. Moreover, we design a graph structure module based on the idea of GraIL

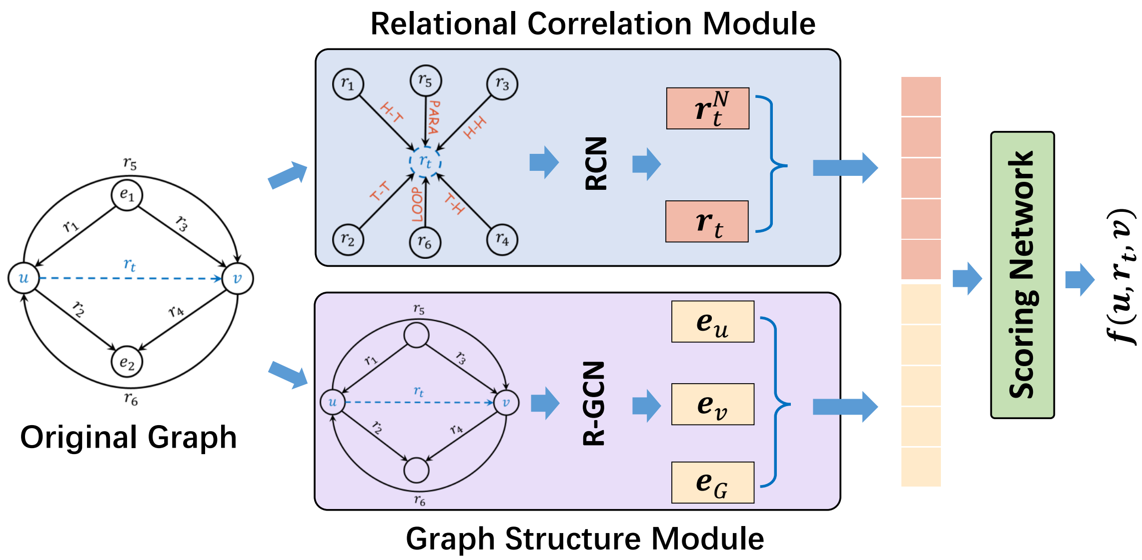

Figure 2: An overview of TACT. TACT consists of two modules: the relational correlation module and the graph structure module. We use a scoring network to score a triple based on the output of the two modules.

(Teru, Denis, and Hamilton 2020) to take advantage of graph structure information. TACT organizes the two modules in a unified framework to perform inductive link prediction. Figure 2 gives an overview of TACT.

Modeling Correlations Between Relations

To model semantic correlations between relations, we consider the correlations in two aspects:

(a) Correlation patterns: The correlations between any two relations are highly correlated to their topological structures in knowledge graphs.

(b) Correlation coefficients: We use correlation coefficients to represent the degree of semantic correlations between any two relations.

Relational Correlation Graph To model the correlation patterns between any two relations, we categorizes all relation pairs into seven topological patterns. As illustrated in Figure 3, the topological patterns are "head-to-tail", "tail-to-tail", "head-to-head", "tail-to-head", "parallel", "loop" and "not connected". We define the corresponding correlation patterns as "H-T", "T-T", "H-H", "T-H", "PARA", "LOOP" and "NC", respectively. For example, we denote $(r_1, \text{H-T}, r_2)$ as the correlation between r_1 and r_2 is the "H-T" pattern for r_2, which indicates that r_1 and r_2 are connected in a head-to-tail manner. (r_1, PARA, r_2) indicates that the two relations are connected by the same head entity and tail entity, and (r_1, LOOP, r_2) indicates that the two relations form a loop in a local graph. We prove that the number of topological patterns between any two relations are at most *seven* in the supplementary.

Based on the definition of different correlation patterns, we can convert the original graph to a Relational Correlation Graph (RCG), where the nodes represent the relations and the edges indicate the correlation patterns between any two relations in the original knowledge graph. Figure 3 shows the topological patterns between any two relations and the corresponding RCGs. Notice that for the topological pattern that two relations are not connected, its corresponding RCG consists of two isolated nodes.

Relational Correlation Network Based on the RCG, we propose a Relational Correlation Network (RCN) to model the importance of different correlation patterns for inductive link prediction. RCN consists of two parts: the correlation pattern part and the correlation coefficient part. The correlation pattern part considers the influence of different topological structures between any two relations. Moreover, the correlation coefficient part aims at learning the degree of different correlations.

For an edge with relation r_t, we can divide all its adjacent edges in the RCG into six groups by the topological patterns "H-T", "T-T", "H-H", "T-H", "PARA" and "LOOP", respectively. Notice that the topological pattern "NC" is not considered as it means the edges (relations) are not connected in the knowledge graph. For the six groups, we use six linear transformations to learn the different semantic correlations corresponding to the topological patterns. To differentiate the degree of different correlations for the relation r_t, we further use attention networks to learn the correlation coefficients for all the correlations.

Specifically, we aggregate all the correlations for the relation r_t to get the neighborhood embedding in a local graph, which is denoted by $\mathbf{r}_t^N$.

$$\mathbf{r}_t^N = \frac{1}{6}\sum_{p=1}^{6}(\mathbf{N}_t^p \circ \mathbf{\Lambda}_t^p)\mathbf{R}\mathbf{W}^p \tag{1}$$

where $\mathbf{W}^p \in \mathbb{R}^{d\times d}$ is the weight parameter, $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}|\times d}$ denotes the embedding of all relations. Suppose the embedding of r_i is $\mathbf{r}_i \in \mathbb{R}^{1\times d}$, then $\mathbf{R}_{[i,:]} = \mathbf{r}_i$ where $\mathbf{R}_{[i,:]}$ denotes the

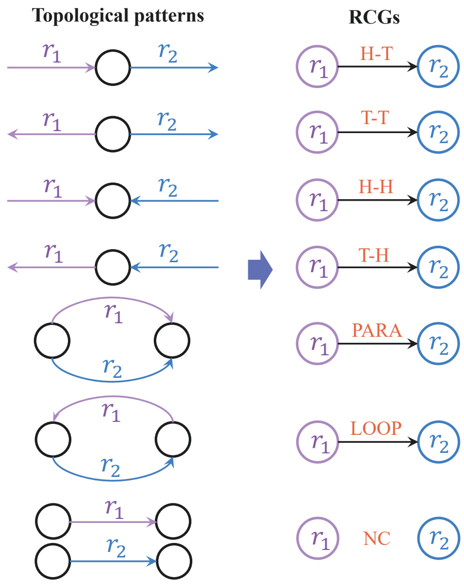

Figure 3: An illustration of the topological patterns between two relations and the corresponding RCGs. For the topological pattern where two relations are not connected, its corresponding RCG consists of two isolated nodes.

i-th slice along the first dimension. $\mathbf{N}_t^p \in \mathbb{R}^{1\times|\mathcal{R}|}$ is the indicator vector where the entry $[\mathbf{N}_t^p]_i = 1$ if r_i and r_t are connected in the p-th topological pattern, otherwise $[\mathbf{N}_t^p]_i = 0$. $\mathbf{\Lambda}_t^p \in \mathbb{R}^{1\times|\mathcal{R}|}$ is the weight parameter which indicates the degree of different correlations for the relation r_t in the p-th correlation pattern. Note that, we restrict $[\mathbf{\Lambda}_t^p]_i \geq 0$ and $\sum_{i=1}^{|\mathcal{R}|}[\mathbf{\Lambda}_t^p]_i = 1$.

Furthermore, we concatenate $\mathbf{r}_t$ and $\mathbf{r}_t^N$ to get the final embedding $\mathbf{r}_t^F$.

$$\mathbf{r}_t^F = \sigma([\mathbf{r}_t \oplus \mathbf{r}_t^N]\mathbf{H}) \tag{2}$$

where $\mathbf{H} \in \mathbb{R}^{2d\times d}$ is the weight parameters, and σ is a activation function, such as $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. We call the module that models semantic correlations between relations as the *relational correlation module*, and $\mathbf{r}_t^F$ is the final output of the module.

Modeling Graph Structures

For a triple (u, r_t, v), the local graph around it contains the information about how the triple connected with its neighborhoods. To take advantage of the local graph structural information, we use a graph structural network to embed local graphs into vectors based on GraIL (Teru, Denis, and Hamilton 2020). To model the graph structure around the triple (u, r_t, v), we perform the following steps: (1) subgraph extraction; (2) node labeling; (3) graph embedding. Notice that nodes represent entities in knowledge graphs.

Subgraph Extraction For a triple (u, r_t, v), we first extract the enclosing subgraph around the target nodes u and v (Teru, Denis, and Hamilton 2020). The enclosing subgraph between nodes u and v is given by the following steps. First, we compute the neighbors $\mathcal{N}_k(u)$ and $\mathcal{N}_k(v)$ of the two nodes u and v, respectively, where k denotes the max distance of neighbors. Second, we take an intersection of $\mathcal{N}_k(u)$ and $\mathcal{N}_k(v)$ to get $\mathcal{N}_k(u) \cap \mathcal{N}_k(v)$. Third, we compute the enclosing subgraph $\mathcal{G}(u, r_t, v)$ by pruning nodes of $\mathcal{N}_k(u) \cap \mathcal{N}_k(v)$ that are isolated or at a distance larger than k from either node u or v.

Node Labeling Afterwards, we label the nodes in the extracted enclosing subgraph following (Teru, Denis, and Hamilton 2020). We label each node i in the subgraph around nodes u and v with the tuple $(d(i, u), d(i, v))$, where $d(i, u)$ denotes the shortest distance between nodes i and u without counting any path through v (likewise for $d(i, v)$). This captures the topological position of each node with respect to the target nodes. The two target nodes u and v are uniquely labeled $(0, 1)$ and $(1, 0)$. The node features are defined as $[\text{one-hot}(d(i, u)) \oplus \text{one-hot}(d(i, v))]$, where $\text{one-hot}(p) \in \mathbb{R}^{1\times d}$ represent the one-hot vector that only the p-th entry is 1, where d represent the dimension of embeddings.

Graph Embedding After node labeling of the enclosing subgraph, the nodes in the subgraph have initial embeddings. We use R-GCN (Schlichtkrull et al. 2018) to learn the embeddings on the extracted enclosing subgraph $\mathcal{G}(u, r_t, v)$.

$$\mathbf{e}_i^{(k+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{e}_j^{(k)}\mathbf{W}_r^{(k)} + \mathbf{e}_i^{(k)}\mathbf{W}_0^{(k)}\right)$$

where $\mathbf{e}_i^{(k)}$ denotes the embedding of entity e_i of the k-th layer in the R-GCN. $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in \mathcal{R}$. $c_{i,r} = |\mathcal{N}_i^r|$ is a normalization constant. $\mathbf{W}_r^{(k)} \in \mathbb{R}^{d\times d}(r \in \mathcal{R}), \mathbf{W}_0^{(k)} \in \mathbb{R}^{d\times d}$ are the weight parameters. $\sigma(\cdot)$ is a activation function, such as the $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$.

Suppose that the number of layers in R-GCN is L, we calculate the embedding of the subgraph $\mathcal{G}(u, r_t, v)$ as

$$\mathbf{e}_{\mathcal{G}(u,r_t,v)}^{(L)} = \frac{1}{|\mathcal{V}_{\mathcal{G}(u,r_t,v)}|}\sum_{i\in\mathcal{V}_{\mathcal{G}(u,r_t,v)}}\mathbf{e}_i^{(L)},$$

where $\mathcal{V}_{\mathcal{G}(u,r_t,v)}$ denotes the set of nodes in graph $\mathcal{G}(u, r_t, v)$. Combining the target nodes and subgraph embedding, the structural information is represented by the vector $\mathbf{e}_S \in \mathbb{R}^{1\times 3d}$,

$$\mathbf{e}_S = \mathbf{e}_{\mathcal{G}(u,r_t,v)}^{(L)} \oplus \mathbf{e}_u^{(L)} \oplus \mathbf{e}_v^{(L)}$$

We call the module that models graph structures as the *graph structure module*, and $\mathbf{e}_S$ is the final output of the module.

The Framework of TACT

Scoring Network The relational correlation module and graph structure module output the embedding vectors $\mathbf{r}_t^F$ and $\mathbf{e}_S$, respectively. To organize the two modules in a unified framework, we design a scoring network to combine the outputs of the two modules and get the score for a given triple (u, r_t, v). The score function $f(u, r_t, v)$ is defined as

$$f(u, r_t, v) = [\mathbf{r}_t^F \oplus \mathbf{e}_S]\mathbf{W}_S$$

	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
GraIL	94.32	94.18	85.80	92.72	84.69	90.57	91.68	94.46	86.05	92.62	93.34	87.50
TACT-base	**98.11**	97.11	88.34	**97.25**	87.36	**94.31**	**97.42**	98.09	94.00	94.44	93.98	94.93
TACT	96.15	**97.95**	**90.58**	96.15	**88.73**	94.20	97.10	**98.30**	**94.87**	**96.58**	**95.70**	**96.12**

Table 1: AUC-PR results on the inductive benchmark datasets extracted from WN18RR, FB15k-237 and NELL-995. The results of Neural LP, DURM and GraIL are taken from the paper (Teru, Denis, and Hamilton 2020).

where $\mathbf{W}_S \in \mathbb{R}^{4d\times 1}$ is a weight parameter.

Loss Function We perform negative sampling and train the model to score positive triples higher than the negative triples using a noise-contrastive hinge loss following (Bordes et al. 2013). The loss function $\mathcal{L}$ is

$$\mathcal{L} = \sum_{i\in[n],(u,r_t,v)\in\mathcal{G}} \max(0, f(u_i', r_{t,i}', v_i') - f(u, r_t, v) + \gamma)$$

where γ is the margin hyperparameter and $\mathcal{G}$ denotes the set of all triples in the knowledge graph. $(u_i', r_{t,i}', v_i')$ denotes the i-th negative triple of the ground-truth triple (u, r_t, v) and $[n]$ represent the set $\{1, 2, \cdots, n\}$, where n is the number of negative samples for each triple.

Experiments and Analysis

This section is organized as follows. First, we introduce the experimental configurations, including datasets, implementation details, and the baseline model. Second, we show the effectiveness of our proposed approach TACT on several benchmark datasets. Finally, we show the results of ablation studies, case studies, and further experiments. The code of TACT is available on GitHub at https://github.com/MIRALab-USTC/KG-TACT.

Experimental Configurations

Datasets We use the benchmark datasets for inductive link prediction proposed in GraIL (Teru, Denis, and Hamilton 2020), which are derived from WN18RR (Toutanova and Chen 2015), FB15k-237 (Dettmers et al. 2018), and NELL-995 (Xiong, Hoang, and Wang 2017). For inductive link prediction, the train set and the test set should have no overlapping entities. Each knowledge graph of WN18RR, FB15k-237, and NELL-995 induces four versions of inductive datasets with increasing sizes. Details of the datasets are summarized in Table 2.

Training Protocol We use Adam optimizer (Kingma and Ba 2015) with initial learning rate of 0.01 and batch size of 16. We randomly sample two-hop enclosing subgraphs for each triple when training and testing, and use a two-layer GCN to calculate the embeddings of subgraphs. The margins in the loss functions are set to 8, 16, 10 for WN18RR, FB15k-237, NELL-995, respectively. The maximum number of training epochs is set to 10.

		WN18RR			FB15k-237			NELL-995		
		#R	#E	#TR	#R	#E	#TR	#R	#E	#TR
v1	#tn	9	2746	6678	183	2000	5226	14	10915	5540
	#tt	9	922	1991	146	1500	2404	14	225	1034
v2	#tn	10	6954	18968	203	3000	12085	88	2564	10109
	#tt	10	2923	4863	176	2000	5092	79	4937	5521
v3	#tn	11	12078	32150	218	4000	22394	142	4647	20117
	#tt	11	5084	7470	187	3000	9137	122	4921	9668
v4	#tn	9	3861	9842	222	5000	33916	77	2092	9289
	#tt	9	7208	15157	204	3500	14554	61	3294	8520

Table 2: Statistics of inductive benchmarks. We use #E and #R and #TR to denote the number of entities, relations, and triples, respectively. We use #tn to denote the train set and #tt to denote the test set.

The Baseline Model To evaluate the effectiveness of our proposed relational correlation module, we propose a baseline called TACT-base, which scores a triple (u, r_t, v) only relying on the output of the relational correlation module. That is, the score function of TACT-base is

$$f_{\text{base}}(u, r_t, v) = \mathbf{r}_t^F \mathbf{W}_{base}$$

where $\mathbf{W}_{\text{base}} \in \mathbb{R}^{d\times 1}$ is a weight parameter.

Inductive Link Prediction

We evaluate the models on both classification and ranking metrics. For both metrics, we compare our method to several state-of-the-art methods, including Neural LP (Yang, Yang, and Cohen 2017), DRUM (Sadeghian et al. 2019), and GraIL (Teru, Denis, and Hamilton 2020).

Classification Metric We use area under the precision-recall curve (AUC-PR) as the classification metric following GraIL (Teru, Denis, and Hamilton 2020). We replace the head or tail of every test triple with a random entity to sample the corresponding negative triple. Then we score the positive triples with an equal number of negative triples to calculate AUC-PR following GraIL (Teru, Denis, and Hamilton 2020). We run each experiment five times with different random seeds and report the mean results.

Table 1 shows the AUC-PR results for inductive link prediction. Our baseline model TACT-base outperforms the inductive baselines on all the datasets. As TACT-base totally

	WN18RR				FB15k-237				NELL-995			
	v1		v4		v1		v4		v1		v4	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
Neural LP	.615	.548	.351	.195	.086	.073	.057	.041	.175	.050	.087	.032
DRUM	.450	.277	.429	.260	.080	.054	.049	.026	.238	.170	.092	.018
GraIL	.851	.749	.746	.610	.054	.016	.056	.017	.337	.146	.072	.017
TACT-base	.990	.983	.981	.966	.804	.700	**.593**	**.409**	.877	.756	.304	.171
TACT	**.995**	**.995**	**.988**	**.982**	**.830**	**.741**	.575	.378	**.880**	**.776**	**.571**	**.444**

Table 3: MRR and H@1 results on the inductive benchmark datasets extracted from WN18RR, FB15k-237, and NELL-995. We reimplement the three baselines Neural LP, DRUM, and GraIL under the inductive relation prediction protocol, with all the hyperparameters keeping the same with their original papers for a fair comparison.

	WN18RR				FB15k-237				NELL-995			
	v1		v4		v1		v4		v1		v4	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
TACT w/o RA	.875	.793	.806	.670	.118	.041	.112	.023	.473	.322	.071	.023
TACT w/o RC	.953	.913	.965	.935	.666	.539	.525	.335	.669	.390	.508	.317
TACT	**.995**	**.995**	**.988**	**.982**	**.830**	**.741**	**.575**	**.378**	**.880**	**.776**	**.571**	**.444**

Table 4: Ablation results the inductive benchmark datasets extracted from WN18RR, FB15k-237, and NELL-995. "TACT w/o RA" represent the baseline that omits the relation aggregation in TACT. "TACT w/o RC" represent the baseline that performs relation aggregation without modeling correlations between relations in TACT.

relies on the relational correlation module to perform link prediction, the results demonstrate the effectiveness of our proposed model for inductive link prediction. TACT further improves the performance of TACT-base and get around 4% improvement against GraIL on most datasets. The experiments demonstrate the effectiveness of modeling topology-aware correlations between relations in TACT for the inductive link prediction task.

Ranking Metric We further evaluate the model for the inductive *relation* prediction to verify the effectiveness of modeling relational correlations in TACT. Inductive relation prediction aims at predicting the target relation between the given head and tail entities. Specifically, for a given relation prediction $(u, ?, v)$ in the test set, we rank the ground-truth relation r against all other candidate relations. Following the standard procedure in prior work (Bordes et al. 2013), we use the filtered setting, which does not take any existing valid triples into account at ranking. We choose Mean Reciprocal Rank (MRR) and Hits at N (H@N) as the evaluation metrics. As the baselines are evaluated in the setting of head/tail prediction, we reimplement Neural LP (Yang, Yang, and Cohen 2017), DRUM (Sadeghian et al. 2019), and GraIL (Teru, Denis, and Hamilton 2020) under our ranking setting. For a fair comparison, we keep the hyperparameters the same with their original papers. Following GraIL, we run each experiment five times with different random seeds and report the mean results.

Table 3 shows the results of MRR and H@1 on v1 and v4 of WN18RR, FB15k-237, and NELL-995. More results about v2 and v3 of WN18RR, FB15k-237, and NELL-995 are listed in the supplementary. As we can see, TACT significantly outperforms rule learning based methods (Yang, Yang, and Cohen 2017; Sadeghian et al. 2019) and GraIL (Teru, Denis, and Hamilton 2020) for the inductive relation prediction. The improvements on FB15k-237 and NELL-995 are more significant than WN18RR. FB15k-237 and NELL-995 contain much more relations than WN18RR, thus semantic correlations between relations are more complex in FB15k-237 and NELL-995. The experiments show that GraIL has difficulty in modeling relational semantics in the condition that the number of relations is large. In contrast, TACT can model the complex patterns of relations through exploiting correlations between relations in knowledge graphs. TACT-base also can significantly outperform existing state-of-the-art methods.

	WN18RR(v1)	NELL-995(v1)	FB15k-237(v1)
#FRE	0.763	0.201	0.470
TACT	**0.995**	**0.830**	**0.880**

Table 5: The MRR results of the frequency-based method and our proposed TACT. We use #FRE to represent the frequency-based method.

Ablation Studies

In this part, we conduct ablation studies on the relation aggregation and modeling correlations between relations.

TACT w/o RA In our proposed method, we aggregate the relation embedding $\mathbf{r}_t$ and neighborhood embedding $\mathbf{r}_t^N$ to get the final relation embedding $\mathbf{r}_t^F$. We omit the aggregation of neighborhood embedding, that is, we let the output of the relational correlation module to be $\mathbf{r}_t$. We called this method "TACT w/o RA".

TACT w/o RC Modeling topology-aware correlations between relations is one of our main contributions. We design a baseline that performs relation aggregation without modeling correlations between relations. That is, the baseline reformulates the equation (1) as

$$\mathbf{r}_t^N = \frac{1}{|\mathcal{N}(r_t)|} \sum_{i \in \mathcal{N}(r_t)} \mathbf{r}_i$$

where $\mathcal{N}(r_t)$ represents the set of neighborhood relations of r_t. We call this baseline "TACT w/o RC" for short.

Table 4 shows the results on three benchmark datasets. The experiments demonstrate the effectiveness of modeling topology-aware correlations between relations in TACT. As correlations between relations are common in knowledge graphs, the relation aggregation in "TACT w/o RC" can take advantage of the neighborhood relations, which is helpful for inductive link prediction. Our proposed method further distinguishes the correlation patterns and correlation coefficients between relations, which makes the learned embeddings of relations more expressive for inductive link prediction. As we can see, TACT significantly outperforms "TACT w/o RA" and "TACT w/o RC" on all the datasets.

The Frequency-based Method We conduct an experiment by ranking relations according to their frequencies and compare TACT with the frequency-based method on the datasets. For the frequency-based method, the returned rank list for every prediction is the same, which is the rank according to the relation frequencies from high to low in the KG. In other words, the frequency-based method represent a type of data bias in the datasets. As illustrated in Table 5, TACT outperforms the frequency-based method by a large margin on the datasets. The results demonstrate that the effectiveness of TACT is not due to the data bias of relation frequencies in benchmark datasets.

Target relation	Most relevant relations	CP	CC
_member_meronym	*_has_part*	PARA	0.68
	_similar_to	H-H	0.39
	_synset_domain_topic_of	T-H	0.31
_similar_to	*_similar_to*	LOOP	0.40
	_member_meronym	H-H	0.39
	_instance_hypernym	T-T	0.35
head_quartered_in	*television_station_affiliated_with*	H-H	0.52
	head_quartered_in	PARA	0.33
	acquired	T-H	0.30

Table 6: Some relations and their top 3 relevant relations. The relations are taken from WN18RR and NELL-995. We use CP to represent correlation pattern, and use CC to represent correlation coefficient.

	AUC-PR	MRR	Hits@1
GraIL	0.634	0.158	0.048
TACT	**0.915**	**0.406**	**0.140**

Table 7: The results for inductive link prediction on the dataset YAGO3-10.

	WN18RR(v1)	NELL-995(v1)	FB15k-237(v1)
GraIL	0.18 h	0.17 h	0.32 h
TACT	0.22 h	2.53 h	8.97 h

Table 8: The running time of TACT on the datasets.

Further Experiments

Case Studies We select some relations and show the top three relevant relations of them in table 6. Recall that the sum of correlation coefficients for each correlation pattern is 1. The results show that TACT can learn some correct correlation patterns. For example, among all the neighboring relations of "_member_meronym", "_has_part"—which is adjacent to "_member_meronym" in the topological pattern of parallel—gets the largest correlation coefficient, as "_has_part" and "_member_meronym" have similar semantics. We list more examples in the supplementary material.

Results on YAGO3-10 To demonstrate the effectiveness of our proposed method on larger KG with few relations. We conduct experiments on YAGO3-10, which is a subset of YAGO3 (Mahdisoltani, Biega, and Suchanek 2015) and contains 37 relaitons and 123,182 entities. Table 7 show the results for inductive link prediction of GraIL and TACT on YAGO3-10. As we can see, TACT outperforms GraIL by all the metrics, which demonstrate our proposed TACT can effectively deal with larger KG with few relations.

Running Time Table 8 show the running time of GraIL and TACT. TACT would take more time than GraIL for modeling the correlations between relations, but the total running time is still acceptable.

Conclusion

In this paper, we propose a novel inductive reasoning approach called TACT, which can effectively exploit topology-aware correlations between relations for inductive link prediction in knowledge graphs. TACT categorizes all relation pairs into several *topological patterns*, and then use the proposed RCN to learn the importance of the different patterns for inductive link prediction. Experiments demonstrate that our proposed TACT significantly outperforms several existing state-of-the-art methods on benchmark datasets for the inductive link prediction task.

Acknowledgments

We would like to thank all the anonymous reviewers for their insightful comments. This work was supported in part by NSFC (61822604, 61836006, 62021001, U19B2026).

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*, 1247–1250. ACM.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*, 2787–2795. Curran Associates, Inc.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*, 1811–1818. AAAI Press.

Do, K.; Tran, T.; and Venkatesh, S. 2018. Knowledge Graph Embedding with Multiple Relation Projections. In *ICPR*, 332–337. IEEE Computer Society.

Hamaguchi, T.; Oiwa, H.; Shimbo, M.; and Matsumoto, Y. 2017. Knowledge Transfer for Out-of-Knowledge-Base Entities : A Graph Neural Network Approach. In *IJCAI*, 1802–1808. ijcai.org.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge Graph Embedding Based Question Answering. In *WSDM*, 105–113. ACM.

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

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*. OpenReview.net.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research*. www.cidrdb.org.

Muggleton, S. 1992. *Inductive logic programming*. Morgan Kaufmann.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In *NeurIPS*, 15321–15331. Curran Associates, Inc.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, volume 10843, 593–607. Springer.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*. OpenReview.net.

Teru, K. K.; Denis, E.; and Hamilton, W. L. 2020. Inductive Relation Prediction by Subgraph Reasoning. In *ICML*, volume 119, 9448–9457. PMLR.

Toutanova, K.; and Chen, D. 2015. Observed Versus Latent Features for Knowledge Base and Text Inference. In *The Workshop on CVSC*. ACL - Association for Computational Linguistics.

Tran, H. D.; Stepanova, D.; Gad-Elrab, M. H.; Lisi, F. A.; and Weikum, G. 2016. Towards Nonmonotonic Relational Learning from Knowledge Graphs. In *Inductive Logic Programming - 26th International Conference*, volume 10326 of *Lecture Notes in Computer Science*, 94–107. Springer.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*. OpenReview.net.

Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; and Guo, M. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *CIKM*, 417–426. Association for Computing Machinery.

Wang, P.; Han, J.; Li, C.; and Pan, R. 2019. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *AAAI*, 7152–7159. AAAI Press.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *EMNLP*, 564–573. Association for Computational Linguistics.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *NIPS*, volume 30, 2319–2328. Curran Associates, Inc.

Zhang, Z.; Cai, J.; and Wang, J. 2020. Duality-Induced Regularizer for Tensor Factorization Based Knowledge Graph Completion. In *NeurIPS*, 21604–21615. Curran Associates, Inc.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020a. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In *AAAI*, 3065–3072. AAAI Press.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *AAAI*, 1441–1451. Association for Computational Linguistics.

Zhang, Z.; Zhuang, F.; Zhu, H.; Shi, Z.; Xiong, H.; and He, Q. 2020b. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion. In *AAAI*, 9612–9619. AAAI Press.

Zhu, J.; Jia, Y.; Xu, J.; Qiao, J.; and Cheng, X. 2018. Modeling the Correlations of Relations for Knowledge Graph Embedding. *J. Comput. Sci. Technol.* 33(2): 323–334.