TDN: Triplet Distributor Network for Knowledge Graph Completion

Jiapu Wang, Boyue Wang, Junbin Gao, Xiaoyan Li, Yongli Hu, and Baocai Yin

Abstract—Conventional Knowledge Graph Completion (KGC) methods typically map entities and relations to a unified space through the shared mapping matrix, and then interact with entities and relations to infer the missing items in the knowledge graph. Although this shared mapping matrix considers the suitability of all triplets, it neglects the specificity of each triplet. To solve this problem, we dynamically learn one information distributor for each triplet to exchange its specific information. In this paper, we propose a novel Triplet Distributor Network (TDN) for the knowledge graph completion task. Specifically, we adaptively learn one Triplet Distributor (TD) for each triplet to assist the interaction between the entity and relation. Furthermore, on the basis of TD, we creatively design the information exchange layer to dynamically propagate the information of the entity and relation, thus mutually enhancing entity and relation representations. Except for several commonly-used knowledge graph datasets, we still implement the link prediction task on the social-relational and medical datasets to test the proposed method. Experimental results demonstrate that the proposed method performs better than existing state-of-the-art KGC methods. The source codes of this paper are available at https://github.com/TDN for Knowledge Graph Completion.git.

Index Terms—Triplet Distributor Network, Knowledge Graph Completion, Knowledge Graph Embedding, Attention Mechanism.

1 Introduction

Knowledge Graph (KG) is one primary form of knowledge base where knowledge is often stored as graph-structured data. Typically, KG is a directed graph whose nodes represent the head/tail entities and edges denote the relations from the head entities to tail entities. The structured knowledge in KG is usually expressed as triplets ($\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$), e.g., ($Albert\ Einstein$, $expert\_in$, $Physics$), where $Albert\ Einstein$ and $Physics$ respectively represent the head entity $\mathbf{h}$ and the tail entity $\mathbf{t}$, and $expert\_in$ is the relation $\mathbf{r}$. Using KG has achieved much success in the artificial intelligence and data mining areas, including information retrieval [1], semantic analysis [2] and dialog systems [3], to name a few.

Despite existing knowledge graphs, e.g., Freebase [4], WordNet [5], NELL [6] and YAGO [7], contain large amounts of triplets, they are far from complete due to the constant emergence of new knowledge, which hinders the practical application effects. This problem directly drives and inspires the Knowledge Graph Completion (KGC) task. The target of the KGC task is to predict the missing entity or relation according to the known information. An illustration about KGC is displayed in Fig. 2, in which we use existing triplets to infer the missing link represented by the dashed line, i.e., ($Albert\ Einstein$, $expert\_in$, $Physics$) is inferred according to existing triplets (solid lines) such as

• Jiapu Wang, Boyue Wang, Xiaoyan Li, Yongli Hu, and Baocai Yin are with Beijing Municipal Key Lab of Multimedia and Intelligent Software Technology, Beijing Artificial Intelligence Institute, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. Junbin Gao is with the Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, NSW 2006, Australia.
E-mail: jpwang@emails.bjut.edu.cn, {wby, xiaoyan.li, huyongli, ybc}@bjut.edu.cn, junbin.gao@sydney.edu.au (Corresponding author: Boyue Wang).

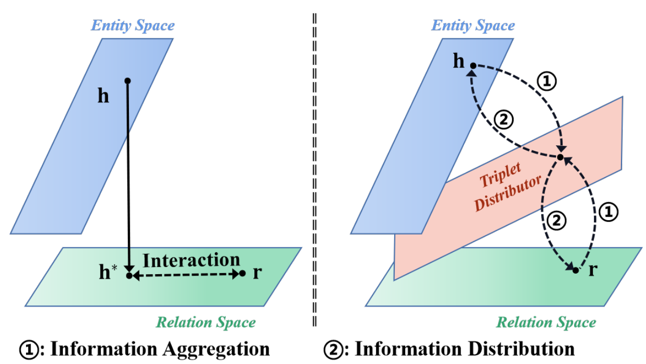


Fig. 1. The motivation of conventional knowledge graph completion methods (left) and the proposed triplet distributor network based method (right). In addition, the serial number indicates the order of the information flow.

($Albert\ Einstein$, $winner\_of$, $Nobel\ Prize\ in\ Physics$) and ($Nobel\ Prize\ in\ Physics$, $award\_in$, $Physics$).

According to Microsoft Concept Graph [8] (shown in TABLE 1), the entity/relation contains multiple concepts, and these concepts typically correspond to different relations/entities. Thus, the information in relations can effectively help entity representation learning, and vice versa. Most KGC methods, such as TransE [9], TransH [10] and TuckER [11], assume entities and relations to be in the same space and use deep learning or specific functions to achieve effective information interaction between entities and relations. However, TransR [12] demonstrates that some entities may be close to each other in the entity space, but they are far apart in the relation space, which makes a common space insufficient for modeling.

Thus, TransR [12] models embeddings of entities and relations in different spaces, and then maps entities from

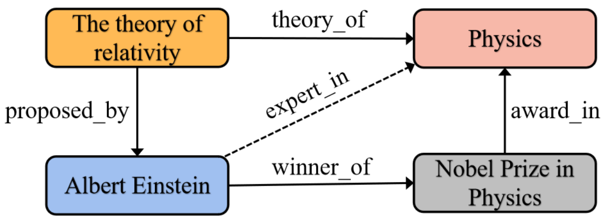


Fig. 2. The brief illustration of knowledge graph completion tasks, containing the existing triplets (solid lines) and the inferred ones (dashed lines).

TABLE 1
Microsoft Concept Graph. "Scores" reflects the frequency of each entity concept.

Entities	Concept	Scores
Apple	fruit	0.415
Apple	company	0.286
Apple	brand	0.050
Orange	fruit	0.350
Orange	citrus fruit	0.251
Orange	bright color	0.084
Cambridge	university	0.347
Cambridge	city	0.214
Cambridge	institution	0.053

the entity space to the corresponding relation space through the shared mapping matrices, enabling the prediction of missing items through their interaction. It should be noted that entities and relations only fully express their semantics in specific spaces, while TransR neglects the semantic loss problem due to the mapping operation from the entity space to the corresponding relation space. Meanwhile, the above-shared mapping matrices in TransR only consider the suitability of all triplets and neglect the specificity of each triplet.

In this paper, we propose a novel **T**riplet **D**istributor **N**etwork (TDN) to tackle above drawbacks. Specifically, instead of mapping entities and relations to the unified space using the shared mapping matrices like TransR [12], the TDN adaptively learns a **T**riplet **D**istributor (TD) for each triplet to transfer the information about entities and relations in different spaces. Fig. 1 briefly displays above principle.

The proposed method is built on a deep learning architecture, which mainly incorporates a **H**ybrid **A**ttention **M**odule (HAM) and multiple **T**riplet **D**istributor **N**etwork **M**odules (TDNM). In each TDNM, the information exchange between entity and relation is achieved through the TD, and TD contains two important functions. One is the information aggregation that dynamically aggregates the information of the entity and relation to the TD. The other is the information distribution that propagates the information of the TD to the entity and relation, thus to symmetrically refine the entity and relation representations for the further representation learning. Furthermore, the tailored TD is embedded in different layers to hierarchically facilitate the learning of the entity and relation representations. Finally, we display the brief framework of our proposed method in Fig. 3.

The main contributions are gathered in the following:

- To our best knowledge, it is the first work that effectively exchanges the information between entities and relations meanwhile keeping them still located in different spaces;
- We creatively design the **T**riplet **D**istributor **N**etwork **M**odule (TDNM), which dynamically facilitates the information aggregation and distribution between entities and relations based on **T**riplet **D**istributor (TD);
- We introduce the **H**ybrid **A**ttention **M**odule (HAM) to further capture the intrinsic correlation between entities and relations;
- Except for testing on the commonly-used knowledge graphs, we also evaluate our proposed method on the social-relational and medical datasets for the link prediction task, and experimental results show the competitive performance.

The rest part of the paper is organized as follows. The research about knowledge graph completion is firstly reviewed in Section 2. In Section 3, we describe the proposed triplet distributor in detail. Then, the whole framework of our proposed method is presented in Section 4. In Section 5, we verify the effectiveness of our proposed method through a set of experiments, and present and analyze the experimental results. Finally, the main contributions are concluded in Section 6.

2 Related Works

Knowledge graph completion has been one popular research in the areas of artificial intelligence and data mining [13]. We roughly divide existing KGC methods into four categories in the following.

2.1 Notion

Given one knowledge graph $\mathcal{G} = \{\mathcal{T} \mid \mathcal{E},\ \mathcal{R}\}$, we have entity set $\mathcal{E}$, relation set $\mathcal{R}$ and triplet set $\mathcal{T}$. Here, the knowledge graph $\mathcal{G}$ consists of triplets expressed by $(\mathbf{h},\ \mathbf{r},\ \mathbf{t}) \in \mathcal{T}$, where $\mathbf{h} \in \mathcal{E}$ means the head entity, $\mathbf{r} \in \mathcal{R}$ denotes the relation, and $\mathbf{t} \in \mathcal{E}$ is the tail entity.

2.2 Translation based Methods

The translation based methods regard the relation as a translation from the head entity to the tail entity, i.e. $head + relation = tail$. Classic methods in such category include TransE [9] and RotatE [14].

TransE [9] is the most widely-used KGC method, which initially defines the translation mechanism $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ to estimate the confidence of $(\mathbf{h}, \mathbf{r}, \mathbf{t})$. Its score function is set as $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{\ell_1/\ell_2}$. On the basis of TransE, **TransH** [10] learns a normal vector of the relation hyperplane to linearly map the entity into the space of relation. In order to capture multiple aspects of an entity, **TransR** [12] constructs one mapping matrix for each relation and then also linearly projects the entity to the space of relation. For each triplet, **TransD** [15] dynamically constructs two mapping matrices

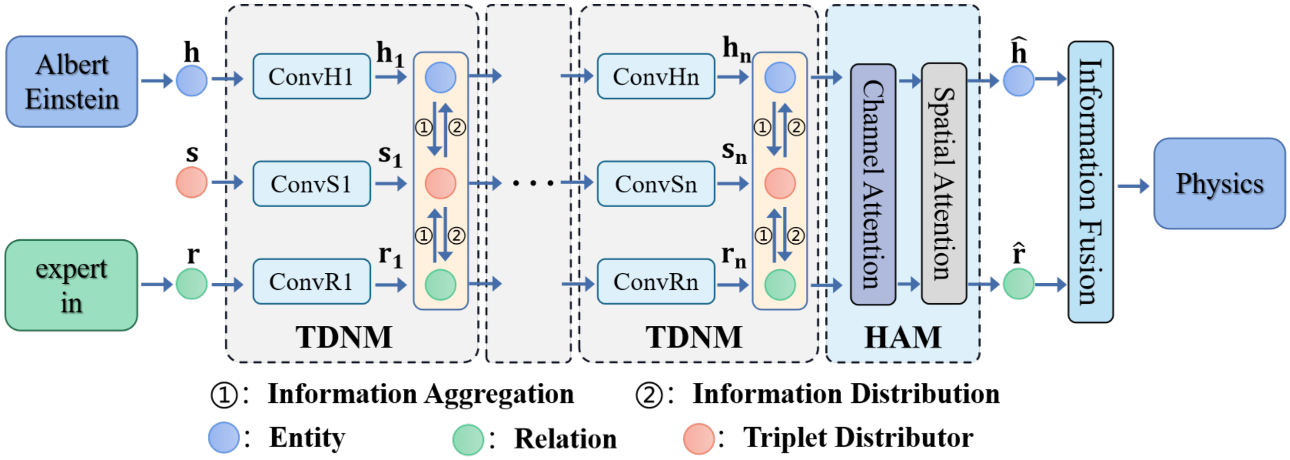


Fig. 3. The architecture of the proposed **T**riplet **D**istributor **N**etwork (TDN) for knowledge graph completion, where *Albert Einstein* and *expert in* represent the head entity and relation, and *Physics* denotes the predicted tail entity. Specifically, TDN mainly consists of multiple **T**riplet **D**istributor **N**etwork **M**odules (TDNM) and a **H**ybrid **A**ttention **M**odule (HAM), and the light gold box represents the process of exchanging information between entities and relations based on **T**riplet **D**istributor (TD).

to respectively project the head and tail entities to the space of relation. Moreover, **TorusE** [16] further improves TransE by embedding the knowledge graph into a Lie group, solving the unchangeable ratio problems of negative sampling and regularization. **RotatE** [14] embeds the knowledge graph into the complex vector space, which regards the relation as a rotation between the head entity and tail entity to model various relation patterns. Recently, **MuRP** [17] model knowledge graphs into a hyperbolic space, and **MuRMP** [18] simultaneously embeds knowledge graphs into Euclidean, hyperspherical and hyperbolic spaces to capture multiple kinds of structures of the knowledge graph.

2.3 Semantic Matching based Methods

Such methods use a similarity-based score function to evaluate the probabilities of triplets. The classic methods include DistMult [19] and TuckER [11].

RESCAL [20] initially proposes the semantic matching method. RESCAL implements the bilinear product between the head entity $\mathbf{h}$ and the tail entity $\mathbf{t}$, and sets a full rank matrix $\mathbf{M}_r$ for the relation. The score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^T \mathbf{M}_r \mathbf{t}$. On the basis of RESCAL, **DistMult** [19] restricts a diagonal matrix $\mathbf{M}_r$ to significantly decrease the scale of relation parameters. **ComplEx** [21] augments the expressive power of DistMult by extending the entity and relation embeddings into the complex space. Moreover, **A2N** [22] takes DistMult as the decoder, and exploits the neighbor information to update the embeddings of entities and relations. **TuckER** [11] introduces tensor to realize the interaction between entity and relation, and further applies the Tucker decomposition to overcome the over-parameterization problem in such methods. **LowFER** [23] introduces an efficient and constraint-free method, which facilitates the better fusion of entities and relations through factorized bilinear pooling. To handle the hierarchical data of the knowledge graph, **ATTH** [24] embeds the knowledge graph into the hyperbolic space to capture the hierarchical structure of knowledge graph. The recent **McRL** [25] mines the complex conceptual information hidden in triplets to solve the multi-concept issue. Moreover, **GIE** [26] interactively learns the spatial structures in the Euclidean, hypersphere and hyperbolic spaces to capture the complex structures of knowledge graphs accurately.

2.4 Convolutional Neural Network based Methods

These methods leverage convolutional neural networks (CNN) to mine more feature interactions between individual embeddings of one triplet, including ConvE [27], RGHAT [28], InteractE [29] and so on.

ConvE [27] directly sends the reshaped head entity and relation into CNN to achieve the tail entity prediction. **HypER** [30] treats the relation as the convolution kernel and the entity as the feature map to reinforce the interaction between them. **ParamE** [31] regards the CNN parameters as the embeddings of relation to fuse advantages of previous TransE and ConvE. **RGHAT** [28] exploits the neighbor information to update the entity and relation embeddings from the relation-level and entity-level attention mechanism. **InteractE** [29] is an enhanced version of ConvE that enhances the expressiveness of the model through deeper interactions. **M-DCN** [32] utilizes a multi-scale dynamic convolutional network to increase the feature interactions between entities and relations. The recent **Conv3D** [33] employs 3D convolutions to capture deeper feature interactions in KG.

2.5 Graph Convolutional Neural Network based Methods

These methods introduce the graph convolutional neural networks (GCN) into knowledge graph completion research to capture the graph structure information, including R-GCN [34] and DRGI [35].

R-GCN [34] firstly introduces the GCN into the knowledge graph completion researches, which effectively utilizes the connections between nodes and relations in the knowledge graph. Based on ConvE, **SACN** [36] applies the graph structural information, node attributes and relation types to

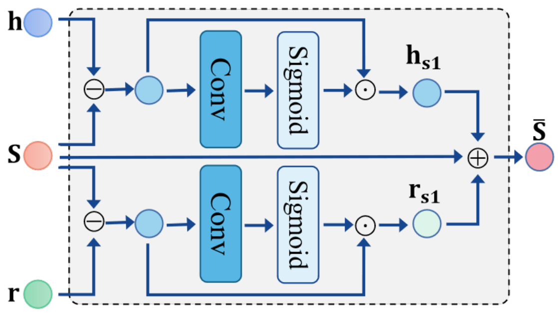


Fig. 4. The illustration of information aggregation module. Specifically, the triplet distributor $\mathbf{s}$ dynamically aggregates the information of the entity $\mathbf{h}$ and relation $\mathbf{r}$ to obtain the enhanced $\bar{\mathbf{s}}$.

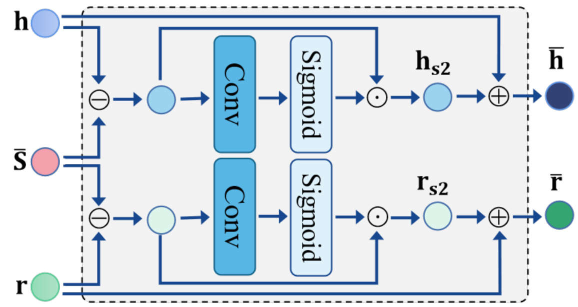


Fig. 5. The illustration of information distribution module. Specifically, the enhanced triplet distributor $\bar{\mathbf{s}}$ dynamically distributes the information to the entity and relation to obtain the enhanced entity representation $\bar{\mathbf{h}}$ and relation representation $\bar{\mathbf{r}}$.

optimize the nodes representations. Meanwhile, **CompGCN** [37] is based on R-GCN, which jointly embeds both entities and relations by a variety of composition operations from KGE techniques. **KE-GCN** [38] combines the advantages of GCN and the advanced KGE methods. Recently, **DRGI** [35] simultaneously captures the complete structural and semantic information from the knowledge graph by building two identical adaptive relational graph attention networks. **HKGN** [39] introduces hypernetworks to mitigate the explosive growth in the number of heterogeneous parameters.

Most aforementioned existing methods directly/indirectly fuse the information of entities and relations through linear/nonlinear mapping operations, thus neglecting two crucial aspects. One is the difficulty of information exchange due to the fact that entities and relations are located in different spaces; the second is the problem of semantic loss due to mapping operations.

3 Triplet Distributor

Entities and relations are often located in different spaces, thus hindering the information exchange between them. To overcome this problem, we adaptively build the **T**riplet **D**istributor (TD) for each triplet to facilitate the information exchange between the entity and relation through a dual information propagation mechanism, which mainly contains the information aggregation operation and the information distribution operation. The architecture of the information aggregation and distribution are respectively illustrated in Fig. 4 and Fig. 5.

Information Aggregation: Given the entity representation $\mathbf{h}$ and the relation representation $\mathbf{r}$, we adaptively propagate the complementary information of the entity and relation to the triplet distributor $\mathbf{s}$ through two gating functions. The architecture of the information aggregation module is shown in Fig. 4.

The information aggregation operation completes the complementary information transfer process by differencing between the entity $\mathbf{h}$ and TD $\mathbf{s}$, and the relation $\mathbf{r}$ and TD $\mathbf{s}$, respectively. So, two gating functions can be defined as follows,

$$\begin{aligned}\mathbf{h_{s1}} &= (\mathbf{h}-\mathbf{s}) \odot [\sigma(f^{1\times1}(\mathbf{h}-\mathbf{s}))] \\ \mathbf{r_{s1}} &= (\mathbf{r}-\mathbf{s}) \odot [\sigma(f^{1\times1}(\mathbf{r}-\mathbf{s}))],\end{aligned} \tag{1}$$

where $f^{1\times1}$ denotes the 1×1 convolution operation; σ represents the *sigmoid* activation function; $\odot$ represents the element-wise multiplication.

Then, to further aggregate more meaningful information to the triplet distributor, we introduce the residual network to complete the information as,

$$\bar{\mathbf{s}} = \mathbf{s} \oplus \mathbf{h_{s1}} \oplus \mathbf{r_{s1}}, \tag{2}$$

where $\oplus$ represents the summation operation. With such an information aggregation mechanism, the TD $\bar{\mathbf{s}}$ contains all information about the entity, relation and itself. Thus, the representation ability of the TD $\bar{\mathbf{s}}$ is enhanced.

Information Distribution: Then we distribute the information of the TD $\bar{\mathbf{s}}$ to refine the entity and relation features, respectively. As shown in Fig. 5, two gating functions are exploited to dynamically propagate the complementary information to refine the entity and relation features. Such process can be expressed as:

$$\begin{aligned}\mathbf{h_{s2}} &= (\mathbf{h}-\bar{\mathbf{s}}) \odot [\sigma(f^{1\times1}(\mathbf{h}-\bar{\mathbf{s}}))] \\ \mathbf{r_{s2}} &= (\mathbf{r}-\bar{\mathbf{s}}) \odot [\sigma(f^{1\times1}(\mathbf{r}-\bar{\mathbf{s}}))],\end{aligned} \tag{3}$$

where f denotes a convolution operation and the superscript 1×1 means the size of the convolutional filter; σ represents the sigmoid activation function; $\odot$ is the element-wise multiplication.

The residual network mechanism is employed again to complement the information to the entity and relation features as follows,

$$\begin{aligned}\bar{\mathbf{h}} &= \mathbf{h} \oplus \mathbf{h_{s2}} \\ \bar{\mathbf{r}} &= \mathbf{r} \oplus \mathbf{r_{s2}}.\end{aligned} \tag{4}$$

For convenience, the above whole process can be abbreviated as an abstract form,

$$\bar{\mathbf{h}},\ \bar{\mathbf{r}},\ \bar{\mathbf{s}} = \mathrm{TD}(\mathbf{h},\ \mathbf{r},\ \mathbf{s}). \tag{5}$$

4 Methodology

In this section, we creatively propose the **T**riplet **D**istributor **N**etwork (TDN) for the knowledge graph completion task. As displayed in Fig. 3, the core of our proposed method TDN mainly contains three components:

- **Triplet distributor network module** facilitates the information aggregation and distribution between entities and relations;
- **Hybrid attention module** further captures the intrinsic correlation between entities and relations;
- **Information fusion module** fuses the information of entities and relations to predict the missing items.

4.1 Triplet Distributor Network Module

Entities and relations may often be located in different spaces, which hinders the information exchange between them. In order to solve this problem, we adaptively build the auxiliary **T**riplet **D**istributor (TD) for each triplet. On the basis of the TD, we propose the **T**riplet **D**istributor **N**etwork **M**odule (TDNM), which is composed of an information exchange layer and three independent 3 × 3 convolutional layers. Specifically, the entity $\mathbf{h}$ and the relation $\mathbf{r}$ are fed into the designed TDNM module to fully exchange information about the entity and relation through the TD.

Each TDNM module contains the entity, relation, and triplet distributor branches. The triplet distributor branch takes a zero-tensor $\mathbf{s}$ as the initial input and hierarchically aggregates the information of the entity and relation. As shown in the TDNM box in Fig. 3, every branch alternatively executes the convolutional layer and information exchange layer to enhance the corresponding feature representation. Specifically, the information exchange layer dynamically propagates the information of the entity and relation through the TD. This process can be denoted as follows,

$$\begin{aligned}\mathbf{h_1},\ \mathbf{r_1},\ \mathbf{s_1} &= \mathrm{TD}(f^{3\times3}(\mathbf{h}),\ f^{3\times3}(\mathbf{r}),\ f^{3\times3}(\mathbf{s}))\\ &= \mathrm{TDNM}(\mathbf{h},\ \mathbf{r},\ \mathbf{s}),\end{aligned} \tag{6}$$

where $\mathbf{h_1}, \mathbf{r_1}$ and $\mathbf{s_1}$ are the corresponding enhanced feature representations of $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{s}$, respectively; TDNM is the proposed triplet distributor network module.

Then, these enhanced features are transferred into a series of TDNM modules to further learn the high-level feature representations of the entity, relation and triplet distributor. This process can be denoted as follows,

$$\mathbf{h_n},\ \mathbf{r_n},\ \mathbf{s_n} = (n-1)\ *\ \mathrm{TDNM}(\mathbf{h_1},\ \mathbf{r_1},\ \mathbf{s_1}), \tag{7}$$

where $\mathbf{h_n}$, $\mathbf{r_n}$ and $\mathbf{s_n}$ are the corresponding enhanced features of $\mathbf{h_1}$, $\mathbf{r_1}$ and $\mathbf{s_1}$, respectively; and n denotes the number of TDNM module.

4.2 Hybrid Attention Module

To further capture the intrinsic correlation between entities and relations, we build the **H**ybrid **A**ttention **M**odule (HAM), motivated by the strategy in [40]. HAM constructs the channel attention sub-module and the spatial attention sub-module, where the channel attention sub-module explores the inter-connection between entities and relations and the spatial attention sub-module explores the intra-connection in entities and relations.

Assuming the head entity $\mathbf{h_n} \in \mathbb{R}^{u\times v}$ and the relation $\mathbf{r_n} \in \mathbb{R}^{u\times v}$, we obtain the feature $\mathbf{F}$ by performing a concatenation operation on the head entity and relation along the channel axis. The process can be represented as follows,

$$\mathbf{F} = \mathrm{Concat}(\mathbf{h_n},\ \mathbf{r_n}), \tag{8}$$

where $\mathbf{F} \in \mathbb{R}^{2\times u\times v}$, u and v are hyper-parameters; $\mathrm{Concat}(\cdot,\cdot)$ represents the concatenation operation. Then feature $\mathbf{F}$ can be taken as the input of the HAM module.

HAM sequentially learns one $1D$ channel attention map $\mathbf{M_c} \in \mathbb{R}^{2\times1\times1}$ and one $2D$ spatial attention map $\mathbf{M_s} \in \mathbb{R}^{1\times u\times v}$ by the channel attention sub-module and the spatial attention sub-module. Above process can be represented as follows,

$$\hat{\mathbf{F}} = \mathbf{M_s} \odot (\mathbf{M_c} \odot \mathbf{F}), \tag{9}$$

where $\hat{\mathbf{F}}$ represents the final refined output. During the element-wise multiplication, attention values are propagated accordingly: spatial attention values and channel attention values propagate along the spatial and channel dimensions, respectively [40].

The channel attention sub-module uses the global max-pooling mechanism, average-pooling mechanism and fully connected layers to explore the inter-connection relationship between entities and relations. This sub-module is described as follows,

$$\begin{aligned}\mathbf{M_c} &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(\mathbf{F})) + \mathrm{MLP}(\mathrm{MaxPool}(\mathbf{F})))\\ &= \sigma(\mathbf{W_1}(\mathbf{W_0}\mathbf{F}^c_{\mathrm{avg}}) + \mathbf{W_1}(\mathbf{W_0}\mathbf{F}^c_{\mathrm{max}})),\end{aligned} \tag{10}$$

where $\mathbf{F}^c_{\mathrm{avg}} \in \mathbb{R}^{2\times1\times1}$ and $\mathbf{F}^c_{\mathrm{max}} \in \mathbb{R}^{2\times1\times1}$ are feature maps obtained from the average-pooling and max-pooling mechanisms; σ represents the sigmoid activation function; $\mathbf{W_0}$ and $\mathbf{W_1}$ denote the weighted matrices.

For the spatial attention sub-module, we exploit the inter-spatial relationship by applying max-pooling and average-pooling mechanisms again along the channel axis. This sub-module can be described as follows,

$$\begin{aligned}\mathbf{M_s} &= \sigma(f^{3\times3}(\mathrm{Concat}(\mathrm{AvgPool}(\mathbf{F}),\ \mathrm{MaxPool}(\mathbf{F}))))\\ &= \sigma(f^{3\times3}(\mathrm{Concat}(\mathbf{F}^s_{avg},\ \mathbf{F}^s_{max}))),\end{aligned} \tag{11}$$

where $\mathbf{F}^s_{\mathrm{avg}} \in \mathbb{R}^{1\times u\times v}$ and $\mathbf{F}^s_{\mathrm{max}} \in \mathbb{R}^{1\times u\times v}$ are feature maps after the average-pooling and max-pooling mechanisms; $f^{3\times3}$ represents the 3×3 convolutional layer.

Then, we take the split operation on feature map $\hat{\mathbf{F}}$ along the channel axis to obtain the enhanced entity and relation embeddings. Finally, we transfer the enhanced entity and relation embeddings into the fully connected layers to obtain the final updated entity representation $\hat{\mathbf{h}}$ and relation representation $\hat{\mathbf{r}}$.

For convenience, we represent the above process as follows,

$$\hat{\mathbf{h}},\ \hat{\mathbf{r}} = \mathrm{HAM}(\mathbf{h_n},\ \mathbf{r_n}), \tag{12}$$

where HAM represents the hybrid attention module.

4.3 Information Fusion Module

To fuse the information of the entity and relation to predict the missing item, we build the information fusion module, which can be substituted by existing KGC methods, such as TransE [9], DistMult [19] and ConvE [27].

Here, we take TransE as an example. We get the missing entity representation $\hat{\mathbf{t}}$ by combining above two representations $\hat{\mathbf{h}}$ and $\hat{\mathbf{r}}$ as,

$$\hat{\mathbf{t}} = \hat{\mathbf{h}} + \hat{\mathbf{r}}. \tag{13}$$

TABLE 2
Statistic information of whole datasets.

Datasets	WN18RR	FB15k-237	WN18	FB15k	Kinship	UMLS
Entities	40,943	14,541	40,943	14,951	104	135
Relations	11	237	18	1,345	25	46
Training	86,835	271,115	141,442	483,142	8,544	5,216
Validation	3,034	17,535	5,000	50,000	1,068	652
Test	3,134	20,466	5,000	59,071	1,074	661

For convenience, the whole proposed method can be simply represented as:

$$\hat{\mathbf{t}} = \mathrm{TDN}(\mathbf{h},\ \mathbf{r}), \tag{14}$$

where TDN represents the proposed triplet distributor network.

Finally, we define the score function $f_r(\mathbf{h},\ \mathbf{t})$ as the following form,

$$f_r(\mathbf{h},\ \mathbf{t}) = - \parallel TDN(\mathbf{h},\ \mathbf{r}) - \mathbf{t} \parallel^2 . \tag{15}$$

4.4 Objective Function

As for the objective function, we minimize the cross-entropy loss with the uniform negative sampling. Here the negative examples for a triplet $(\mathbf{h},\ \mathbf{r},\ \mathbf{t})$ are sampled uniformly from all possible triplets by perturbing the tail entity,

$$\mathcal{L} = \sum_{\mathbf{t}' \in \mathcal{U}(\mathcal{V})} \log(1 + \exp(y_{(\mathbf{h},\mathbf{r},\mathbf{t}')} f_r(\mathbf{h},\ \mathbf{t}'))), \tag{16}$$

where $\mathcal{U}(\mathcal{V})$ represents the candidate set for the tail entity, and $y_{(\mathbf{h},\mathbf{r},\mathbf{t}')} = \begin{cases} -1, & \text{if } \mathbf{t}' = \mathbf{t} \\ 1, & \text{otherwise} \end{cases}$ marks the positive and negative samples. In order to intuitively exhibit the whole operating procedure of our proposed method, we summarize the training process in Algorithm 1.

Algorithm 1: Triplet Distributor Network for Knowledge Graph Completion.

Input: The head entity $\mathbf{h}$, the relation $\mathbf{r}$, and the initial network parameters θ;
for epoch $= 0 : M - 1$ **do**
  for TDNM $= 1 : N_{TDNM}$ **do**
    Aggregating the information of the entity and relation to the triplet distributor through (2);
    Distributing the information of the new triplet distributor to refine the entity and relation representations through (4);
  end for
  Capturing the intrinsic correlation between the entity and relation through the HAM module (12);
  Obtaining the predicted tail entity $\hat{\mathbf{t}}$ through (13);
  Calculating the score of the triplet through (15);
  Optimizing the entire network by minimizing the objective function (16);
end for
return The predicted tail entity.

5 Experiment

In this section, we test the proposed method by conducting the link prediction tasks.

5.1 Datasets

Four commonly-used knowledge graphs and two domain knowledge graphs are chosen to test the proposed method in our experiments. The major statistics of the datasets are summarized in TABLE 2.

- **WN18** [9]: A subset of WordNet has lexical relations between words;
- **FB15k** [9]: A subset of Freebase have many facts about the real world;
- **WN18RR** [27]: A subset from WN18 contains no inverse relations. This dataset is challenge due to the inverse relations in the training set help training so that the triplets in the testing set can be predicted easily;
- **FB15k-237** [41]: A subset from FB15k has no inverse relations similar to WN18RR;
- **Alyawarra Kinship** [42]: Aboriginal Australian kinship knowledge graph has the aboriginal customary law governing social interaction systems referring to kinship in traditional aboriginal cultures;
- **Unified Medical Language Systems (UMLS)** [43]: A medical terminology knowledge graph brings together many health and biomedical vocabularies and standards.

5.2 Baselines

Our proposed method TDN is compared to a set of related KGC methods,

- **TransE** [9]: TransE is the first translation based method that models the translation mechanism between entities and relations.
- **DistMult** [19]: DistMult is one classic semantic matching method, which uses a bilinear product to define the score function.
- **TransR** [12]: TransR is the first work that models entities and relations in different spaces.
- **ComplEX** [21]: ComplEX embeds entities and relations into the complex space, which is an enhanced version of DistMult.
- **ConvE** [27]: ConvE is the first convolutional neural network based method, which models the interaction between entities and relations through the 2D convolutional neural network.

TABLE 3
Link prediction results on WN18 and FB15k datasets. The best-performing and the second-best-performing results are in blod and in underline, respectively. † and ¶ denote that the results are taken from [11] and [32], and other results are directly taken from the original papers. In addition, − means the corresponding result is unavailable.

Methods	WN18				FB15k			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TransE (2013)¶	0.454	0.934	0.823	0.089	0.380	0.641	0.472	0.231
DistMult (2015)†	0.822	0.936	0.914	0.728	0.654	0.824	0.733	0.546
TransR (2015)¶	0.605	0.940	0.876	0.335	0.346	0.582	0.404	0.218
ComplEX (2016)†	0.941	0.947	0.945	0.936	0.692	0.840	0.759	0.599
ConvE (2018)†	0.943	0.956	0.946	0.935	0.657	0.831	0.723	0.558
TorusE (2018)	0.947	0.954	0.950	0.943	0.733	0.832	0.771	0.674
HypER (2019)	0.951	0.958	<u>0.955</u>	0.947	0.790	0.885	0.829	0.734
TuckER (2019)†	0.953	0.958	<u>0.955</u>	<u>0.949</u>	0.795	<u>0.892</u>	0.833	0.741
DRGI (2021)	0.948	0.957	0.950	0.943	0.798	**0.896**	0.854	0.746
M-DCN (2022)¶	0.950	0.958	0.954	0.946	0.762	0.879	0.820	0.701
Conv3D (2022)	**0.998**	**0.999**	**0.999**	**0.998**	0.228	0.374	0.246	0.153
TDN (TransE)	<u>0.955</u>	<u>0.959</u>	0.951	0.946	<u>0.828</u>	0.890	<u>0.863</u>	<u>0.788</u>
TDN (DistMult)	<u>0.955</u>	<u>0.959</u>	0.954	<u>0.949</u>	**0.833**	**0.896**	**0.865**	**0.796**
TDN (ConvE)	0.945	0.953	0.948	0.940	0.805	0.885	0.836	0.759
TDN (ComplEX)	0.945	0.953	0.948	0.940	0.735	0.879	0.807	0.688

- **TorusE** [16]: TorusE maps the knowledge graph into a Lie group.
- **SACN** [36]: SACN is an enhanced version of ConvE that introduces graph structure information, node attributes and relation types.
- **A2N** [22]: A2N learns the query-dependent representations of entities based on the graph neural network structure.
- **RotatE** [14]: RotatE treats the relation as a rotation from the head entity to the tail entity in the complex space.
- **HypER** [30]: HypER treats the relation as the convolution kernel and the entity as the feature map to reinforce the interaction between them.
- **Conv3D** [33]: Conv3D is an extension of ConvE, which employs 3D convolutions to capture deeper feature interactions in KG.
- **TuckER** [11]: TuckER is a straightforward and powerful bilinear method based on the Tucker decomposition.
- **InteractE** [29]: InteractE is an extension of ConvE that increases the scale of interactions between entity and relation.
- **ATTH** [24]: ATTH is a classic translation based method, which models the KG into the hyperbolic space.
- **M-DCN** [32]: M-DCN introduces the dynamic convolutional network to increase the interaction between entities and relations.
- **DRGI** [35]: DRGI is an extension of R-GCN that simultaneously captures the semantic information and the complete structure information contained in the knowledge graph.
- **MuRP** [17]: MuRP firstly introduces the hyperbolic embedding into the KGC task.
- **MuRMP** [18]: MuRMP is an extension of ATTH that embeds the knowledge graph into a multi-relational mixed-curvature space with fixed curvature $[1, -1]$.
- **GIE** [26]: GIE is a recent semantic matching method, which captures the complex structure of knowledge through Euclidean, hyperbolic and spherical space.
- **HKGN** [39]: HKGN is an extension of R-GCN, which introduces hypernetworks to mitigate the explosive growth in the number of heterogeneous parameters.

5.3 Link Prediction Metrics

For each test triplet $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, we replace the head or tail entity with all possible entities sampled from the candidate set, then we rank the scores calculated by the score function in Equation (15). Finally, following the evaluation protocol in [9], we report the *filtered* results that filter out all corrupted triplets presented in the knowledge graph.

In order to fairly evaluate the proposed method, four commonly-used metrics are chosen, i.e., MRR, Hits@10, Hits@3 and Hits@1. The higher value indicates the better performance.

5.4 Parameters Setting

On most datasets, we use the Adam algorithm to optimize the objective function of our proposed method, while on FB15k, we use the Adagrad algorithm. To obtain the optimal learning rate and negative samples for different datasets, we vary the learning rate and the number of negative samples per true triplet from $\{0.1, 0.05, 0.01, 0.005, 0.001\}$ and $\{0, 10, 50, 100\}$, respectively. Across all datasets, the optimal learning rate is set to 0.001. Furthermore, we set the number of negative samples to 0 on Kinship, WN, FB15k, and UMLS, and to 50 on WN18RR and FB15k-237. The batch size is set to 128 by default.

We execute a grid search to get the optimal hyper-parameters according to the MRR performance on the

TABLE 4
Link prediction results on WN18RR and FB15k-237 datasets. − means the corresponding result is unavailable. The best-performing and the second-best-performing results are in bolded and underlined. The experimental results of comparison methods are directly taken from the original papers, and § denotes that the results are taken from [24].

Methods	WN18RR				FB15k-237			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TransE (2013)¶	0.182	0.444	0.295	0.027	0.257	0.420	0.284	0.174
DistMult (2014)§	0.430	0.490	0.440	0.390	0.241	0.419	0.263	0.155
TransR (2015)¶	0.401	0.465	0.389	0.345	0.263	0.428	0.267	0.168
ComplEX (2016)§	0.440	0.510	0.460	0.410	0.247	0.428	0.275	0.158
ConvE (2018)§	0.430	0.520	0.440	0.400	0.325	0.501	0.356	0.237
TorusE (2018)	0.464	0.534	0.480	0.429	0.307	0.485	0.337	0.219
SACN (2019)	0.470	0.540	0.480	0.430	0.350	0.540	0.390	0.260
A2N (2019)	0.450	0.510	0.460	0.420	0.317	0.486	0.348	0.232
MuRP (2019)	0.481	0.566	0.495	0.440	0.335	0.518	0.367	0.243
RotatE (2019)§	0.476	0.571	0.492	0.428	0.338	0.533	0.375	0.241
HypER (2019)	0.465	0.522	0.477	0.436	0.341	0.520	0.376	0.252
TuckER (2019)	0.450	0.510	0.460	0.420	0.317	0.486	0.348	0.232
InteractE (2020)	0.460	0.523	–	0.430	0.354	0.535	–	0.263
ATTH (2020)§	0.486	0.573	0.499	0.443	0.348	0.540	0.384	0.252
MuRMP (2021)	0.473	0.552	0.485	0.435	0.345	0.542	0.385	0.258
DRGI (2021)	0.479	0.543	0.496	0.445	<u>0.362</u>	0.549	0.399	0.270
GIE (2022)	0.491	0.575	0.505	0.452	<u>0.362</u>	0.552	0.401	0.271
M-DCN (2022)¶	0.475	0.540	0.485	0.440	0.345	0.528	0.380	0.255
Conv3D (2022)	**0.817**	**0.986**	**0.907**	**0.708**	0.200	0.367	0.232	0.114
HKGN (2022)	0.487	0.561	0.505	0.448	**0.365**	0.552	0.402	0.272
TDN (TransE)	<u>0.499</u>	<u>0.579</u>	<u>0.523</u>	<u>0.455</u>	0.358	**0.561**	<u>0.403</u>	<u>0.273</u>
TDN (DistMult)	0.491	0.575	0.512	0.450	0.360	<u>0.556</u>	**0.407**	**0.277**
TDN (ConvE)	0.477	0.550	0.493	0.430	0.348	0.539	0.392	0.258
TDN (ComplEX)	0.481	0.569	0.502	0.439	0.350	0.546	0.395	0.263

TABLE 5
Link prediction results on Kinship and UMLS datasets. The best-performing and the second-best-performing results are in bolded and underlined. $*$ denotes that the results are taken from [35].

Methods	Kinship				UMLS			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TransE (2013)*	0.211	0.470	0.252	0.093	0.615	0.945	0.807	0.391
DistMult (2015)*	0.480	0.708	0.491	0.377	0.164	0.403	0.135	0.061
RotatE (2016)*	0.738	0.954	0.827	0.617	0.822	0.969	0.932	0.703
ConvE (2018)*	0.772	0.950	0.858	0.665	0.836	0.946	0.917	0.764
SACN (2018)*	0.799	0.964	0.878	0.699	0.856	0.985	0.946	0.764
DRGI (2021)*	0.847	0.981	0.915	0.765	0.898	0.988	0.948	0.838
TDN (TransE)	<u>0.858</u>	0.985	0.934	0.775	<u>0.961</u>	**0.999**	0.983	**0.926**
TDN (DistMult)	**0.867**	<u>0.986</u>	**0.942**	<u>0.788</u>	**0.963**	**0.999**	**0.985**	**0.926**
TDN (ConvE)	0.833	0.979	0.913	0.755	0.956	**0.999**	<u>0.984</u>	<u>0.920</u>
TDN (ComplEX)	**0.867**	**0.987**	<u>0.939</u>	**0.790**	0.938	<u>0.997</u>	0.983	0.891

validation dataset. We tune the embedding size of entities and relations on different datasets in a range of $\{50,\ 100,\ 150,\ 200,\ 250\}$, and finally set the embedding size of entities and relations to 200 on all datasets. For each dataset, the number of TDNM modules n is varied in a range of $\{1,\ 2,\ 3,\ 4\}$. Specifically, the number of TDNM modules is set to 1 on UMLS, and 3 on the remaining datasets. The number of filters for each convolution operation in TDNM modules is tuned from $\{3,\ 8,\ 16,\ 32,\ 64,\ 128\}$, and we set the number of filters for each convolution operation in TDNM modules to 3 on UMLS; 16 on WN18; 32 on WN18RR and FB15k-237; 128 on FB15k and Kinship. Moreover, the hyper-parameters u and v in the hybrid attention module are set to 10 and 20, respectively.

The whole experiments in this paper are run on a PyTorch 1.8.1 platform and a workstation equipped with NVIDIA RTX 3090 GPU and 128G RAM.

TABLE 6
The ablation experiment results on WN18RR and FB15k-237 datasets. We mark the better results in bolded font.

Methods	WN18RR				FB15k-237			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TDN w/o TD	0.472	0.554	0.497	0.432	0.321	0.513	0.366	0.240
TDN w/o SA	0.495	0.577	0.508	0.449	0.353	0.558	0.399	0.266
TDN w/o CA	0.494	0.576	0.506	0.451	0.355	0.559	0.396	0.264
TDN w/o HAM	0.491	0.573	0.518	0.449	0.350	0.556	0.395	0.261
TDN (TransE)	**0.499**	**0.579**	**0.523**	**0.455**	**0.358**	**0.561**	**0.403**	**0.273**

TABLE 7
The ablation experiment results on Kinship and UMLS datasets. We mark the better results in bolded font.

Methods	Kinship				UMLS			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TDN w/o TD	0.775	0.975	0.878	0.655	0.891	0.993	0.942	0.828
TDN w/o SA	0.843	0.983	0.925	0.753	0.944	0.996	0.979	0.908
TDN w/o CA	0.833	0.986	0.922	0.736	0.949	0.996	0.977	0.918
TDN w/o HAM	0.842	0.978	0.915	0.753	0.958	0.995	0.980	0.923
TDN (TransE)	**0.858**	**0.985**	**0.934**	**0.775**	**0.961**	**0.999**	**0.983**	**0.926**

5.5 Result Analysis

We exhibit the link prediction experiment results in TABLE 3, TABLE 4 and TABLE 5. In particular, since datasets and comparison methods used in the original papers are usually different, the comparison methods in this paper on different datasets are also slightly different.

From TABLE 3, TABLE 4 and TABLE 5, several interesting phenomena are discovered in the following:

(1) Experimental results verify that our proposed method performs obviously better than other compared methods in most conditions except for the WN18 and WN18RR. Although the proposed method cannot obtain the best performance on WN18, we believe it is competitive to the excellent performance of the compared methods. Moreover, the experimental results on Kinship and UMLS prove that the proposed method is also good at handling domain knowledge. These phenomena clearly illustrate that the triplet distributor allows for the efficient transfer of information about entities and relations, thereby optimizing the embedded representations. Specifically, the triplet distributor can more effectively facilitate the representation learning of entities and relations in domain knowledge graphs with many proper names, such as Kinship and UMLS.

Although the proposed method cannot perform as well as Conv3D [33] on WN18 and WN18RR, it performs far better than Conv3D on FB15k and FB15k-237. The phenomenon clearly proves the robustness of the proposed method.

(2) Specifically, TuckER is an important baseline due to the similar motivation with the proposed method, i.e., learning a tensor to interact with entities and relations. Although TuckER [11] realizes the interaction between entity and relation, it models entity and relation in a common space, which limits the performance of TuckER to some extent. Compared with TuckER, the proposed method obtains the better performance on all datasets under most metrics. For example, the proposed method TDN (TransE) obtains 3.5% and 4.1% improvements under Hits@1 on WN18RR and FB15k-237. The reason is that we model entities and relations in different spaces, and adaptively learn a Triplet Distributor for each triplet to exchange the information about entities and relations. Experimental results clearly demonstrate the effectiveness of the proposed method.

(3) Moreover, TransE, DistMult, ConvE and ComplEx are also important baselines since they act as the decoder for the proposed method. Compared with these baselines, the proposed method obtains a clear and substantial improvement in all metrics across all datasets. Experimental results clearly show that our proposed method learns the optimal entity and relation representations through the triplet distributor and the hybrid attention module.

Specifically, on small scale datasets, such as Kinship and UMLS, the performance of the proposed method is satisfactory using baselines ConvE and ComplEx. However, on large scale datasets, the baseline having the lower complexity performs well, such as TransE. This phenomenon means that a stronger baseline cannot necessarily increase the model performance. We speculate the reason is that the large scale of parameters and the high complexity on large datasets severely limit the model expressiveness ability.

(4) Importantly, TransR is also a key baseline as it is the first work to model entities and relations in different spaces. Compared with TransR, the proposed method obviously performs better on all datasets. The reason is that the proposed method not only captures the suitability of all triplets, but also uncovers the specificity of each triplet.

The above phenomena prove the effectiveness of the triplet distributor and the hybrid attention module.

5.6 Convergence Analysis

We further test the algorithm convergence through a set of experiments on four important datasets, e.g., WN18RR, FB15k-237, Kinship and UMLS. As shown in Fig. 7, we can observe that the proposed method is relatively steady after a certain number of iterations on these datasets. This

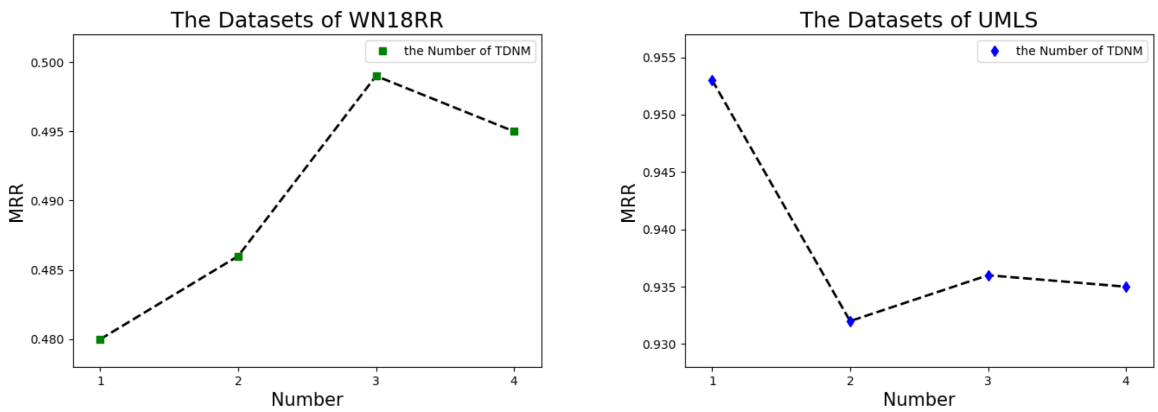


Fig. 6. The impacts of the number of TDNM on WN18RR and UMLS.

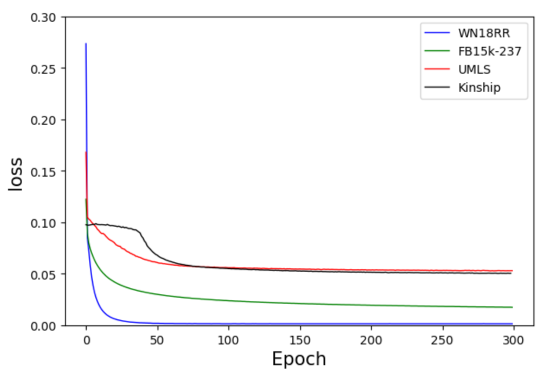


Fig. 7. Convergence performance of the proposed method.

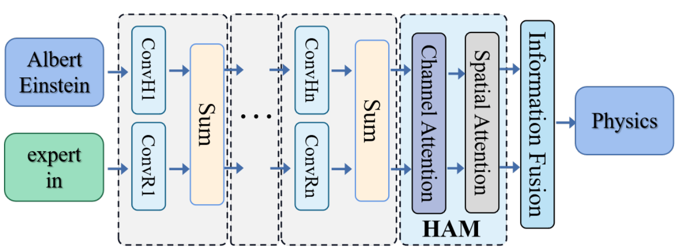


Fig. 8. The framework of the "TDN w/o TD" ablation method, which removes the Triplet Distributor (TD).

phenomenon fully verifies the validity of the proposed method.

5.7 Ablation Experiment

We design four ablation experiments based on the TransE decoder to investigate the influence of the key modules in the proposed method on several important datasets, and the corresponding link prediction results are summarized in TABLE 6 and TABLE 7. The "TDN w/o TD" ablation method removes the triplet distributor and the information exchange layer, and the corresponding framework is shown in Fig. 8. The "TDN w/o HAM" ablation method removes the hybrid attention module (HAM). "TDN w/o CA" ablation method removes the channel attention and "TDN w/o SA" ablation method removes the spatial attention.

From TABLE 6 and TABLE 7, we observe that our proposed method outperforms the "TDN w/o TD" ablation method greatly across all datasets. For example, the proposed method obtains 12% and 3.3% improvements under Hits@1 on Kinship and FB15k-237. The phenomenon verifies that the triplet distributor can effectively solve the difficulty of information exchange between entities and relations, and the information exchange layer can fully facilitate the information aggregation and distribution between them, thus obtaining better link prediction performance.

Compared with the "TDN w/o HAM", "TDN w/o CA" and "TDN w/o SA" ablation methods, the performance of our proposed method obtains certain improvements. This phenomenon demonstrates that the single attention modules (CA, SA) and hybrid attention module (HAM) capture the intrinsic correlation between entities and relations, and HAM performs better than CA and SA.

Thus, these phenomena imply that these key modules in our proposed TDN method contribute to the link prediction performance.

5.8 Impact of the Number of TDNM

To numerically study the impact of the number of triplet distributor network modules (TDNM) on MRR performance of the proposed method TDN (TransE), we conduct comparison experiments on a commonly-used dataset and a domain dataset, such as WN18RR and UMLS. In the comparison experiments, we let the number of TDNM n vary from 1 to 4 and fix all other hyper-parameters, then we observe the MRR values on WN18RR and UMLS.

As shown in Fig. 6, the number of TDNM n affects the link prediction performance significantly. On WN18RR, when n varies from 1 to 3, the MRR value increases steadily and reaches the maximum when it is equal to 3. However, when n becomes even larger, no consistent and dramatic improvements are observed on the WN18RR dataset. On UMLS, when n varies from 1 to 4, the MRR value reaches the maximum when it is equal to 1, and then gradually decreased and eventually stabilized.

The phenomena prove that the appropriate TDNM modules can effectively improve the link prediction perfor-

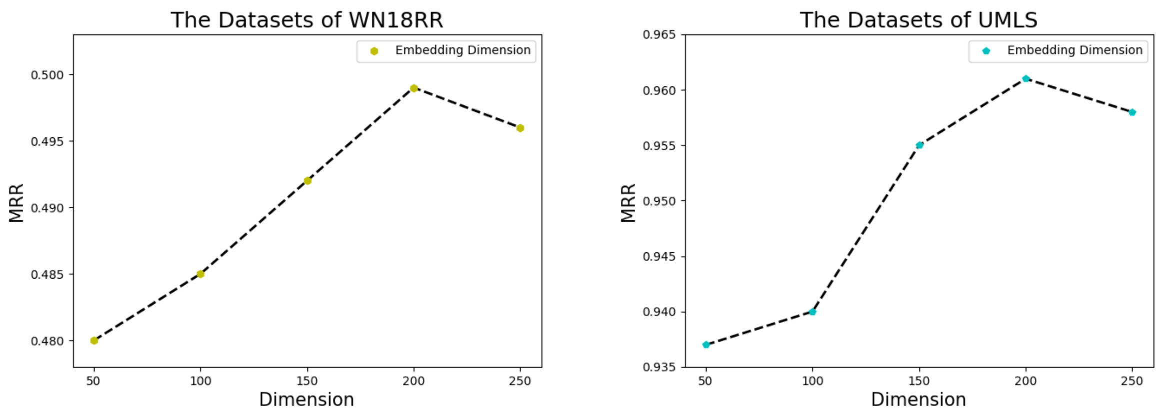


Fig. 9. The impacts of the embedding dimension on WN18RR and UMLS.

TABLE 8
The comparison of parameter scales. – means the corresponding result is unavailable

Methods	RotatE	MuRP	GIE	TDN
WN18RR	40.95M	32.76M	17.65M	10.39M
FB15k-237	29.32M	5.82M	3.41M	5.39M

mance. Still, excessive TDNM modules cannot further improve the link prediction performance due to the over-parameterization problem.

5.9 Impact of the Embedding Dimension

To experimentally study the impact of the embedding dimension on MRR performance of the proposed method TDN (TransE), we choose two important datasets such as WN18RR and UMLS for comparison experiments. In the comparison experiments, we tune the embedding dimension varying in a range of $\{50, 100, 150, 200, 250\}$ for entities and relations, then we observe the MRR values on WN18RR and UMLS.

As shown in Fig. 9, we can observe that the embedding dimension affects the link prediction performance significantly. On both datasets, the value of MRR gradually increases with increasing dimensions in the range of $\{50, 100, 150, 200\}$. However, when the embedding dimension becomes even larger, no consistent and dramatic improvements are observed on both datasets.

The phenomena prove that the appropriate embedding dimension can effectively improve the link prediction performance.

5.10 The scale comparison of parameters

Following the strategy of GIE [26], we choose several mainstream KGC methods to conduct comparative experiments on the scale of parameters in TABLE 8, which shows some advantages of the parameter scale of the proposed method. The phenomenon demonstrates that our proposed method can maintain superior performance with fewer parameters.

6 Conclusion

In this paper, we proposed a novel triplet distributor network for the knowledge graph completion task, which creatively builds the triplet distributor to solve the difficulty of the information exchange due to the fact that entities and relations are often located in different spaces. Specifically, on the basis of the triplet distributor, we designed the information exchange layer to propagate the information of the entity and relation to enhance the entity and relation representations mutually. Moreover, experimental performance on the public datasets proves that our proposed method acquires competitive performance comparing to other state-of-the-art link prediction methods.

In future work, we will further reduce the parameter scale, running time and computation resources to better suit for completing large-scale knowledge graphs. Moreover, we will build a simple and efficient decoder to improve the performance of knowledge graph completion.

Acknowledgements

The research project is partially supported by National Key R&D Program of China (2021ZD0111902); National Natural Science Foundation of China (No. 62272015, U19B2039, U21B2038); R&D Program of Beijing Municipal Education Commission (KZ202210005008).

References

[1] J. Dalton, L. Dietz, and J. Allan, "Entity query feature expansion using knowledge base links," in *ACM SIGIR Conference on Research & Development in Information Retrieval*, 2014, pp. 365–374.
[2] R. I. Goldblatt, "Semantic analysis of orthologic," *Journal of Philosophical Logic*, vol. 3, no. 1-2, pp. 19–35, 1974.
[3] Y. Ma, P. A. Crook, R. Sarikaya, and E. Fosler-Lussier, "Knowledge graph inference for spoken dialog systems," in *International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5346–5350.
[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.
[5] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[6] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, K. Jayant, L. Ni, M. Kathryn, M. Thahir, N. Ndapandula, P. Emmanouil, R. Alan, S. Mehdi, S. Burr, W. Derry, G. Abhinav, C. Xi, S. Abulhair, and W. Joel, "Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[7] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A core of semantic knowledge," in *International Conference on World Wide Web*, 2007, pp. 697–706.

[8] J. Cheng, Z. Wang, J.-R. Wen, J. Yan, and Z. Chen, "Contextual text understanding in distributional semantic space," in *ACM International on Conference on Information and Knowledge Management*, 2015, pp. 133–142.

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2787–2795.

[10] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes." in *AAAI Conference on Artificial Intelligence*, 2014, pp. 1112–1119.

[11] I. Balažević, C. Allen, and T. M. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," in *Empirical Methods in Natural Language Processing*, 2019, pp. 5185–5194.

[12] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI Conference on Artificial Intelligence*, 2015, pp. 2181–2187.

[13] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 2, pp. 1–49, 2021.

[14] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2019, pp. 1–18.

[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 687–696.

[16] T. Ebisu and R. Ichise, "Generalized translation-based embedding of knowledge graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 941–951, 2020.

[17] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," in *Advances in Neural Information Processing Systems*, 2019, pp. 4465–4475.

[18] S. Wang, X. Wei, C. N. D. Santos, Z. Wang, R. M. Nallapati, A. Arnold, B. Xiang, P. S. Yu, and I. F. Cruz, "Mixed-curvature multi-relational graph neural network for knowledge graph completion," in *International Conference on World Wide Web*, 2021, pp. 1761–1771.

[19] B. Yang, W. T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *International Conference on Learning Representations*, 2015, pp. 1–13.

[20] M. Nickel, V. Tresp, and H. P. Kriegel, "A three-way model for collective learning on multi-relational data," in *International Conference on Machine Learning*, 2011, pp. 809–816.

[21] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International Conference on Machine Learning*, 2016, pp. 2071–2080.

[22] T. Bansal, D. C. Juan, S. Ravi, and A. Mccallum, "A2N: Attending to neighbors for knowledge graph inference," in *Annual Meeting of the Association for Computational Linguistics*, 2019, p. 1777–1786.

[23] S. Amin, S. Varanasi, K. A. Dunfield, and G. Neumann, "Lowfer: Low-rank bilinear pooling for link prediction," in *International Conference on Machine Learning*, 2020, pp. 257–268.

[24] I. Chami, A. Wolf, D. C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Annual Meeting of the Association for Computational Linguistics*, 2020, p. 6901–6914.

[25] J. Wang, B. Wang, J. Gao, Y. Hu, and B. Yin, "Multi-concept representation learning for knowledge graph completion," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, pp. 1–19, 2023.

[26] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Geometry interaction knowledge graph embeddings," in *AAAI Conference on Artificial Intelligence*, 2022.

[27] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *AAAI Conference on Artificial Intelligence*, 2018, pp. 1811–1818.

[28] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He, "Relational graph neural network with hierarchical attention for knowledge graph completion," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 9612–9619.

[29] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 3009–3016.

[30] I. Balažević, C. Allen, and T. Hospedales, "Hypernetwork knowledge graph embeddings," in *International Conference on Artificial Neural Networks*, 2019, pp. 553–565.

[31] F. Che, D. Zhang, J. Tao, M. Niu, and B. Zhao, "ParamE: Regarding neural network parameters as relation embeddings for knowledge graph completion," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 2774–2781.

[32] Z. Zhang, Z. Li, H. Liu, and N. N. Xiong, "Multi-scale dynamic convolutional network for knowledge graph embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2335–2347, 2022.

[33] W. Feng, D. Zha, L. Wang, and X. Guo, "Convolutional 3d embedding for knowledge graph completion," in *International Conference on Computer Supported Cooperative Work in Design*, 2022, pp. 1197–1202.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[35] S. Liang, J. Shao, D. Zhang, J. Zhang, and B. Cui, "DRGI: Deep relational graph infomax for knowledge graph completion," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[36] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 3060–3067.

[37] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *International Conference on Learning Representations*, 2020, pp. 1–15.

[38] D. Yu, Y. Yang, R. Zhang, and Y. Wu, "Knowledge embedding based graph convolutional network," in *International Conference on World Wide Web*, 2021, pp. 1619–1628.

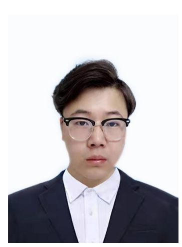
[39] X. Liu, T. Zhu, H. Tan, and R. Zhang, "Heterogeneous graph neural network with hypernetworks for knowledge graph embedding," in *International Semantic Web Conference*, 2022, pp. 284–302.

[40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *European Conference on Computer Vision*, 2018, pp. 3–19.

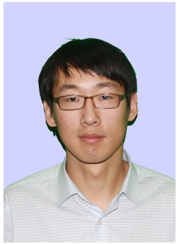
[41] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *The Workshop on Continuous Vector Space Models and their Compositionality*, 2015, pp. 57–66.

[42] X. V. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," in *Empirical Methods in Natural Language Processing*, 2018, pp. 3243–620.

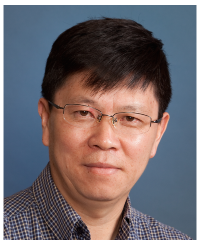
[43] S. Kok and P. Domingos, "Statistical predicate invention," in *International Conference on Machine Learning*, 2007, pp. 433–440.



Jiapu Wang is currently a Ph.D student in the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing. His research interests include knowledge graph completion, computer vision and pattern recognition.



Boyue Wang received his B.Sc. degree in Computer Science from Hebei University of Technology, China in 2012 and obtained PhD from Beijing University of Technology, China in 2018. He is an associate professor in the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing. His current research interests include manifold learning, cross-media, graph learning and knowledge graph.



Junbin Gao graduated from Huazhong University of Science and Technology (HUST), China in 1982 with a BSc. in Computational Mathematics and obtained his PhD. from Dalian University of Technology, China in 1991. He is a Professor of Big Data Analytics in the University of Sydney Business School at the University of Sydney and was a Professor in Computer Science in the School of Computing and Mathematics at Charles Sturt University, Australia. He was a senior lecturer, a lecturer in Computer Science from 2001 to 2005 at the University of New England, Australia. From 1982 to 2001 he was an associate lecturer, lecturer, associate professor, and professor in Department of Mathematics at HUST. His main research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

Xiaoyan Li graduated with a B.Sc degree at Tianjing University in 2014. She received her M.S. degree at Brown University in 2016, and her Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, in 2022. She is currently a postdoctoral fellow at Beijing University of Technology. Her research interests are mainly computer vision, weakly supervised learning, 2D and 3D object detection, semantic segmentation and knowledge graph.

Yongli Hu (M'12) received the Ph.D. degree from the Beijing University of Technology, China, in 2005. He is currently a Professor with the Faculty of Information Technology, Beijing university of Technology. He is also a researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, and with the Beijing Advanced Innovation Center for Future Internet Technology. His research interests include computer vision, pattern recognition, machine learning and multimedia technology.

Baocai Yin received his Ph.D. from Dalian University of Technology in 1993. He is a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. He is a researcher at the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology. He is a member of the China Computer Federation. His research interests include multimedia, multifunctional perception, virtual reality, and computer graphics.