Higher-order Logical Knowledge Representation Learning

Suixue Wang[1] , **Weiliang Huo**[1] , **Shilin Zhang**[2] and **Qingchen Zhang**[3,*]
[1] School of Information and Communication Engineering, Hainan University
[2] College of Intelligence and Computing, Tianjin University
[3] School of Computer Science and Technology, Hainan University
{wangsuixue, wlhuo, zhangqingchen}@hainanu.edu.cn, zhang_shilin_sd@163.com

Abstract

Real-world knowledge graphs abound with higher-order logical relations that simple triples, limited to pairwise connections, fail to represent. Thus, capturing higher-order logical relations involving multiple entities has garnered significant attention. However, existing methods ignore the structural information in higher-order relations. To this end, we propose a higher-order logical knowledge representation learning method, named LORE, which leverages network motifs, the patterns/subgraphs that naturally capture the structural information in graphs, to extract higher-order features and ultimately, learn effective representations of knowledge graphs. Compared to existing approaches, LORE aggregates the attribute features of entities with the extracted higher-order logical relations to form enhanced representations of knowledge graphs. In particular, three aggregators (i.e., Hadamard, Connection, and Summation) are proposed and employed. Extensive experiments have been conducted on seven real-world datasets for two downstream tasks (i.e., entity classification and link prediction). The results show that LORE outperforms baselines significantly and consistently.

1 Introduction

The representation learning of knowledge graphs (KGs) is a long-standing challenge, as KGs serve as foundational components in a wide range of real-world applications, including recommender systems [Zhao *et al.*, 2023; Shokrzadeh *et al.*, 2024], question-answering systems [Wang *et al.*, 2024; Ding *et al.*, 2024], and semantic analysis models [Song *et al.*, 2024b; Zhong *et al.*, 2023; Shan *et al.*, 2023; Sun *et al.*, 2022], to name just a few. It has been demonstrated that in these systems, KGs can be leveraged to improve the overall effectiveness via incorporating structured and machine comprehensible external knowledge [Taunk *et al.*, 2023] and to improve the interpretability of these systems [Li *et al.*, 2024] via providing insights about the underlying decision-making process. KGs structure knowledge as graphs, where nodes represent entities (e.g., concepts or objects) and edges denote relationships between them. KG representation learning aims to encode entities and relations into meaningful representations, capturing their semantics to improve performance in tasks like entity classification and link prediction.

The fundamental unit of data in a KG for representing a single piece of knowledge is a triple, which consists of a subject entity, an object entity, and the relation between them. The usage of triples allows a KG to capture binary relations (i.e., relations involving a pair of entities) precisely. Therefore, various studies have proposed triple-based representation learning methods for KGs [Vashishth *et al.*, 2019; Liao *et al.*, 2021]. However, triples do not explicitly capture those relations that involve multiple entities, i.e., higher-order relations. We illustrate this using the example in Figure 1, where a snapshot of a knowledge graph involving the entities *Steven Allan Spielberg* and *Tom Hanks* is presented. Based on the three triples — *(Steven Allan Spielberg, Cooperate, Tom Hanks)*, *(Steven Allan Spielberg, Director_of, Saving Private Ryan)*, and *(Tom Hanks, Actor_of, Saving Private Ryan)*, three first-order logical relations, i.e., triples, can be established: (1) *Steven Allan Spielberg* has cooperated with *Tom Hanks*; (2) *Steven Allan Spielberg* has directed the film *Saving Private Ryan*; and (3) *Tom Hanks* has played a role in the film *Saving Private Ryan*. However, none of the triples explicitly express the ternary relationship between *Steven Allan Spielberg*, *Tom Hanks*, and *Saving Private Ryan*; linking the semantic meanings of the three triples is required to capture the ternary relationship.

Hence, the development of approaches for explicit modeling of higher-order relations has attracted intensive attention. One popular approach is to extend the triples to multihop paths through graph traversal [Xu *et al.*, 2020]. Considering the importance of different relations encoded via multi-step paths can enhance knowledge graph representation learning and downstream tasks [Donnat *et al.*, 2018; Ranganathan and Barbosa, 2022; Liu *et al.*, 2024; Zong *et al.*, 2024; Zheng *et al.*, 2018]. However, we argue that extracting ***linear*** paths from a knowledge graph is not sufficient to represent the rich semantic meanings of high-order relations in KGs, still with the example in Figure 1. We can formulate two second-order logical relations through path extraction: "Steven Allan Spielberg has collaborated with Tom Hanks, who is an actor in Saving Private Ryan" and "Tom

[*] Corresponding author: Qingchen Zhang.

Hanks has cooperated with Steven Allan Spielberg, who directed the film Saving Private Ryan" (as shown in the middle of Figure 1). In both two higher-order relations, it is unknown how Tom Hanks and Director Spielberg collaborated, although by observing the knowledge graph, one can easily establish the fact that "Tom Hanks played a role in the movie Saving Private Ryan, which is directed by Steven Allan Spielberg". Such information is lost when a model can only extract linear paths.

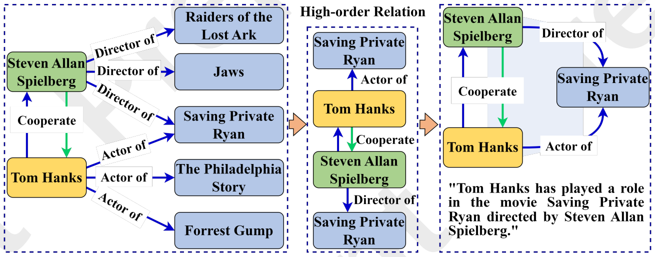


Figure 1: Higher-order knowledge representation.

To address this limitation, we propose to keep the structural information of higher-order relations that involve multiple entities. The main idea is to employ motifs, i.e., subgraphs and patterns that appear frequently in a graph, to represent the structural information of higher-order relations. As shown in Figure 1, the ternary relation between *Steven Allan Spielberg*, *Tom Hanks*, and *Saving Private Ryan* can be readily represented through the triangle network motif. Based on this, we propose a knowledge graph representation learning framework, namely LORE (higher-order **L**ogical kn**O**wledge **R**epresentation l**E**arning), to capture first- and higher-order relational attributes in knowledge graphs simultaneously. In LORE, we consider two aspects of features: (1) the attribute features carried by a knowledge graph, i.e., attribute features of entities and relation types of edges; and (2) the structural features that are represented using heterogeneous motifs. LORE extracts and aggregates these features to enhance the effectiveness of learned graph representations. We further conduct extensive experiments on real-world datasets and evaluate the proposed method on two different tasks: entity classification and link prediction. Experimental results show that the proposed method can significantly improve the knowledge representation effectiveness, providing substantial improvements to downstream tasks. The contribution of this paper can be summarized as follows:

- **Higher-order logical relation formulation**: We tackle the problem of the formulation of higher-order logical relations in the knowledge graph representation learning. Specifically, we propose to employ network motifs to preserve the complex semantic relations within higher-order logical relations in the knowledge graph.
- **Quantifying motif complexity in KGs**: We propose the entity motif degree to quantify the capability of motifs to represent complex relational features and preserve higher-order logical relationships in KGs.
- **Precise knowledge representation**: We propose a knowledge graph representation learning framework, LORE, which jointly considers the first- and higher-order relations for knowledge graph representation learning.
- **Real-world datasets verification**: We conduct extensive experimentats on seven real-world datasets, including AIFB, PPI, MUTAG, BGS, FB15k-237, WN18, and WN18RR, and evaluate the proposed representation learning model on two classification tasks, including entity classification and link prediction. The experimental results confirm the effectiveness of the proposed framework, LORE, with significant improvements over strong baseline methods.

2 Related Work

2.1 Higher-Order Relations in Knowledge Graphs

Relations in knowledge graphs are complex, since two entities can be indirectly related via multiple intermediate nodes. Most existing studies aim to simplify such higher-order logical relations by representing them as multiple simple pairwise relations [Niu *et al.*, 2021; Qiu *et al.*, 2020]. However, accurately representing real-world higher-order logical relations is challenging due to the limitations of the triple representation format. Meanwhile, most of the existing approaches are based on the assumption that the knowledge graph is a set of independent triples, ignoring the structural information in the graph. To address this, the Triple-Context-based Knowledge Embedding model (TCE) proposed by [Shi *et al.*, 2017] represents each triad together with its in-graph context in a unified framework to reflect the structural information in the context of the triad. A deep fact detection model, DEAN, is proposed by [Tu *et al.*, 2023], which captures hidden structural information between facts by comprehensively modeling entities and relationships, thereby identifying obsolete facts in the knowledge graph. Higher-order logical relation occupies a vital position in knowledge graph representation learning and can improve the effectiveness of learning results [Ren *et al.*, 2021]. However, the challenge lies in formulating these higher-order logical relations effectively. Thus, in this work, we employ network motifs to model higher-order logical relations, aiming to enhance the performance of knowledge representation.

2.2 Motif-driven Graph Learning

Network motifs play a vital role in uncovering rich information within graph networks and have diverse applications. For instance, [Wang *et al.*, 2023a] proposes the Motif-based Graph Attention Network (MGSR) for service recommendation, leveraging motif attention mechanisms to capture higher-order information and employing collaborative filtering for predictions. Similarly, [Yu *et al.*, 2020] introduces a motif-dimensional framework that harnesses higher-order structural features to enhance existing network representation learning methods. A novel solution, Dual-level Graph Self-Supervised Pretraining with Motif Discovery (DGPM), is proposed by [Yan *et al.*, 2024]. DGPM autonomously discovers significant graph motifs through an edge pooling module and aligns learned motif similarities with graph kernel-based similarities. This solution addresses challenges such as limited topology learning, human knowledge dependency, and incompetent multi-level interactions faced by self-supervised graph pre-training techniques. In this paper, we use network motifs to formulate higher-order logical relations in knowledge graphs and properly integrate structural relations and attribute features to represent knowledge graphs.

3 Preliminary

3.1 Problem Definition

We represent a knowledge graph (KG) as a directed graph with multi-relations: $\mathcal{KG} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ and $\mathcal{R}$ are the sets of entities and relations, respectively. $\mathcal{T}$ is the triple set consisting of $(h, r, t) \in \{\mathcal{E}, \mathcal{R}, \mathcal{E}\}$, where h and t are the head and tail entity in the relation r, respectively. The task of knowledge graph representation learning is to embed entities and relations in a latent vector space, i.e., $\mathbf{h}, \mathbf{t} \in \mathbb{R}^e$ and $\mathbf{r} \in \mathbb{R}^r$, in such a way that it captures the semantic meanings between them.

3.2 Network Motifs in Knowledge Graphs

Our proposed graph learning framework, LORE, constructs higher-order logical relational features through network motifs. Network motifs refer to those subgraphs/patterns that appear with high frequency in a large graph. Many networks exhibit rich motifs, and effective modelling of them has been proven to be beneficial in exploring graph structures. Figure 2 indicates three-order and four-order motifs. In a knowledge graph, these motifs carry rich relational information features in their topological structures, and hence, can be used to uncover higher-order logical relations between entities. Therefore, we propose the **entity motif degree** for quantifying the capability of a motif in representing relational features. The formal definition is shown as follows:

Definition 1. *Given an entity* e *and a motif* m*, the entity motif degree of* e *w.r.t* m *is defined as the number of subgraphs with the same structure as* m*, whose constituent entities include* e*.*

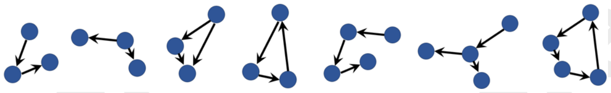


Figure 2: Three-order and four-order motifs.

Figure 3 illustrates the computation of the entity motif degree. Let e be Node 5 and m be the last four-order motif in the middle part of Figure 3. Although m appears four times in the graph (i.e., 1-3-4-2, 3-4-2-1, 4-2-1-3, and 2-1-3-4), none of those subgraphs include Node 5. Hence, the entity motif degree of Node 5 w.r.t. the last four-order motif is 0. On the contrary, the entity motif degree of Node 1 w.r.t. the same motif is 4.

Definition 2. *Given an entity set* E *and a motif set* M*, the entity motif degree matrix is defined as a two-dimensional matrix* $\mathbf{D} \in \mathbb{Z}^{|E| \times |M|}$*:*

$$\mathbf{D} = \left[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{|E|}\right]^{\mathrm{T}}$$

where for any $i \in [1, |E|]$*:*

$$\mathbf{d}_i = \left[d_{i,1}, d_{i,2}, \ldots, d_{i,|M|}\right]$$

Here, $d_{i,j}$ is the entity motif degree of the i-th entity in E w.r.t. j-th motif in M.

Continue with the example in Figure 3, the knowledge graph can be represented by an entity motif degree matrix, where the i-th row represents the entity degree vector of the i-th node w.r.t. the motif set in the middle part of Figure 3, e.g., the first row $\mathbf{d}_1$ indicates the entity motif degree vector of Node 1.

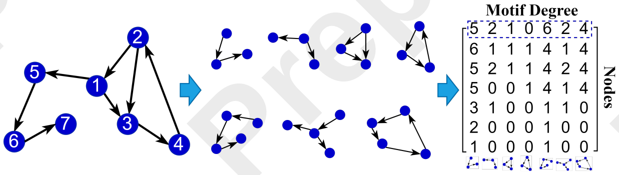


Figure 3: Network motifs and motif degree matrix.

In this paper, we use the entity motif degree matrix to represent the higher-order logical relations carried by a knowledge graph. Each row in the matrix, i.e., each entity motif degree vector, encodes information about the connectivity of the corresponding node in the graph. In the above example, the first row, $\mathbf{D}_1$, has high motif degrees in all columns, reflecting that Node 1 in Figure 3 is a highly-connected node in the graph with rich first- or higher-order logical relations. On the contrary, the fifth row, $\mathbf{D}_5$, is a sparse vector, which reflects the relatively limited logical relations of Node 5 with other nodes.

4 The Design of LORE

This section introduces LORE, a framework for learning knowledge graph representations by jointly modeling entity features and higher-order logical relational features. The overall framework of LORE is shown in Figure 4. LORE mainly consists of three stages: higher-order logical relational features formulation, features representation, and features aggregation. Next, we discuss each of the three stages in detail.

4.1 Higher-order Logical Relation Formulation

We model higher-order logical relations among entities using the entity motif degree matrix introduced in Section 3.2. For each relation $r \in \mathcal{R}$, we generate a subgraph by retaining only edges of type r. The entity motif degree matrix $\mathbf{D}^{(r)}$ for each relation type r captures the higher-order relations among entities with respect to r.

To enhance the representation, we expand the original relation set $\mathcal{R}$ to include edges in both directions (outgoing and incoming). Assuming directed knowledge graphs, edges between two nodes in different directions may have distinct semantic meanings. Therefore, we incorporate the inverse edges, denoted as $\mathcal{R}_{inv}$, allowing the flow of semantic information in both directions. We also include self-loops, which connect an entity to itself, to capture self-referential information. The final relation set, $\mathcal{R}'$, combines the original relations, inverse relations, and self-loops: $\mathcal{R}^{'} = \mathcal{R} \cup \mathcal{R}_{inv} \cup \hat{\mathcal{T}}$.

Note that some relations in $\mathcal{R}'$ may cover fewer entities due to limited motif coverage, leading to long-tail relations.

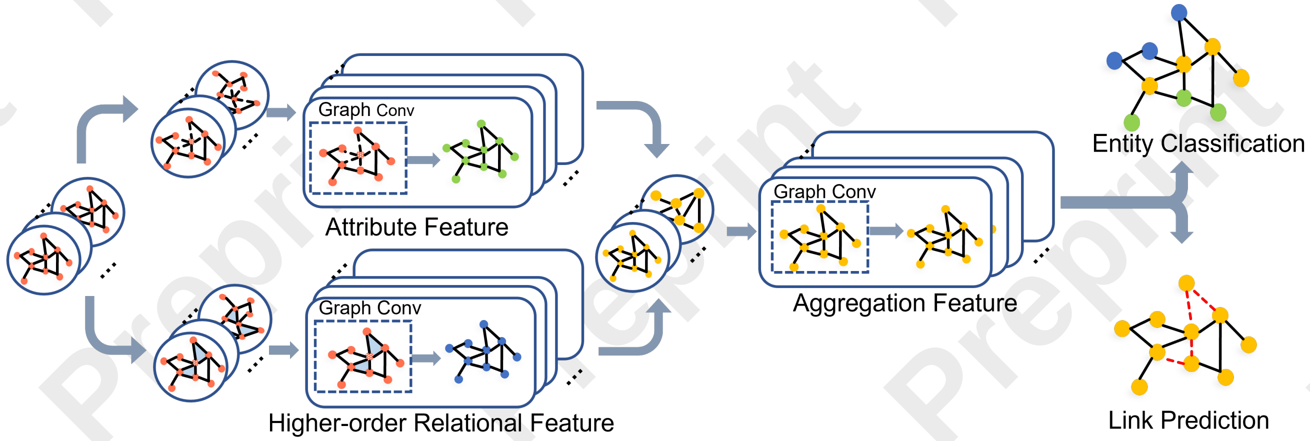


Figure 4: The overall framework of LORE.

To address this, LORE enhances the entity motif degree matrices for sparse relations by randomly adding 1s based on a threshold. This threshold balances matrix sparsity with sufficient information to represent the relation and is empirically set according to the knowledge graph and relation characteristics.

4.2 Feature Representation

Most existing approaches for the representation learning of knowledge graphs rely on the adjacency matrix of a knowledge graph for feature aggregation: the attribute features of direct and multi-hop neighbors are fused to learn the representation of a node. These approaches only capture simple and linear logical relations between nodes. In contrast, LORE can take attribute features and higher-order logical relational features into consideration simultaneously. Let $\mathbf{D}^{(r)}, r \in \mathcal{R}^{'}$ represent the higher-order logical relation for relation type r and $\mathbf{F}^{(r)}$ be the attribute matrix for relation type r. Graph Convolutional Networks (GCN) [Kipf and Welling, 2016] are used to extract useful features from $\mathbf{D}^{(r)}$ and $\mathbf{F}^{(r)}$ and map them into a feature space with the same dimensions. The formula of a single convolutional layer for one relation r is shown as follows:

$$\mathbf{H_d^{l+1}} = \mathbf{\hat{A}H_d^{l}W_d} \tag{1}$$

$$\mathbf{H_f^{l+1}} = \mathbf{\hat{A}H_f^{l}W_f} \tag{2}$$

where $\mathbf{H_d} = [\mathbf{h_d^1}, \mathbf{h_d^2}, \ldots, \mathbf{h_d}^{|\mathcal{E}|}]$, $\mathbf{H_f} = [\mathbf{h_f^1}, \mathbf{h_f^2}, \ldots, \mathbf{h_f}^{|\mathcal{E}|}]$, $\mathbf{\hat{A}} = \mathbf{\tilde{D}}^{-\frac{1}{2}}\mathbf{\tilde{A}\tilde{D}}^{-\frac{1}{2}}$, $\mathbf{\tilde{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{\tilde{D}_{ii}} = \sum_j \mathbf{\tilde{A}_{ij}}$, $\mathbf{W_d}$ and $\mathbf{W_f}$ are the learnable parameters. In the initialization of the formula, $\mathbf{H_d^1} = \mathbf{\hat{A}D}^{(r)}\mathbf{W_d}$, $\mathbf{H_f^1} = \mathbf{\hat{A}F}^{(r)}\mathbf{W_f}$. The entity update formula for all relations is shown as follows:

$$\mathbf{h_i^{l+1}} = \sigma(\sum_{r \in \mathcal{R}^{'}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{|\mathcal{N}_i^r|} \mathbf{W_r^{(l)} h_j^l}) \tag{3}$$

Here, $|\mathcal{N}_i^r|$ indicates the number of the neighbors of entity i, σ is the activation fuction, $\mathbf{W_r^{(l)}}$ is the learnable parameters in the l-th layer. Both features use the same node update method.

Due to a large number of relations in knowledge graphs, there will be a large number of parameters to represent higher-order logical relational features and attribute features. In addition, the overfitting problem also exists. In order to address these two problems, we use the basis decomposition to decompose $\mathbf{W_d}$ and $\mathbf{W_f}$ as follows:

$$\mathbf{W_d} = \sum_{b=1}^{B} a_{db}\mathbf{V_{db}} \tag{4}$$

$$\mathbf{W_f} = \sum_{b=1}^{B} a_{fb}\mathbf{V_{fb}} \tag{5}$$

where a_{db}, a_{fb}, $\mathbf{V_{db}}$, and $\mathbf{V_{fb}}$ are learnable parameters. B denotes the hyperparameter. According to the basis decomposition, the number of parameters to be learned can be reduced to $\frac{B \times dim_d^{(l+1)} \times dim_d^{(l)} + |\mathcal{R}^{'}| \times B}{|\mathcal{R}^{'}| \times dim_d^{(l+1)} \times dim_d^{(l)}} + \frac{B \times dim_a^{(l+1)} \times dim_f^{(l)} + |\mathcal{R}^{'}| \times B}{|\mathcal{R}^{'}| \times dim_f^{(l+1)} \times dim_f^{(l)}}$. Here, $|\mathcal{R}^{'}|$ is the number of the relation, $\mathbf{V_{db}} \in \mathbb{Z}^{dim_d^{(l+1)} \times dim_d^{(l)}}$ and $\mathbf{V_{fb}} \in \mathbb{Z}^{dim_f^{(l+1)} \times dim_f^{(l)}}$. The optimization of the basis matrices $\mathbf{V_{db}}$ and $\mathbf{V_{fb}}$ is shared universally across both common and uncommon relations, so the shared optimization parameters may effectively prevent the occurrence of overfitting on uncommon relations.

4.3 Feature Aggregation

In this phase, the extracted higher-order logical relational features and attribute features from previous stages are merged to obtain a comprehensive feature representation of entities. In LORE, we consider three different aggregation methods: Hadamard, Summation, and Connection aggregations.

Hadamard: Hadamard aggregation is a binary operation, which multiplies the higher-order logical relational features representation $\mathbf{H_d^1}$ and attribute features representation $\mathbf{H_f^1}$. The Hadamard aggregation matrix requires elements to multiply with each other. The calculation equation is shown as follows:

$$\mathbf{H_{agg}} = \mathbf{H_f^1} \odot \mathbf{H_d^1} \tag{6}$$

$$\mathbf{H_{agg}}[\mathbf{i},\mathbf{j}] = \mathbf{H^1_f}[\mathbf{i},\mathbf{j}] \times \mathbf{H^1_d}[\mathbf{i},\mathbf{j}] \tag{7}$$

The dimension of the Hadamard aggregation result is the same as the dimension of the original feature matrices. However, the shortcoming of this aggregator is that it may lead to the loss of information. For example, if $\mathbf{H^1_d}[\mathbf{i},\mathbf{j}]$ is relatively small (e.g., close to 0), the final product, however, will be small even if $\mathbf{H^1_f}[\mathbf{i},\mathbf{j}]$ has a relatively large value. As a result, the feature information in $\mathbf{H^1_f}[\mathbf{i},\mathbf{j}]$ will be lost during the aggregation. In general, we observe that this aggregator may result in the dilution of dominant features.

Connection: Connection refers to the concatenation of the matrices [Wang *et al.*, 2023c]. It is the process of integrating two features into a new matrix. The calculation equation is shown as follows:

$$\mathbf{H_{agg}} = [\mathbf{H^1_d} : \mathbf{H^1_f}] \tag{8}$$

The aggregation dimension of the result is higher than the matrix dimension of $\mathbf{H^1_d}$ and $\mathbf{H^1_f}$, which will make the subsequent iterative training process more time-consuming. Meanwhile, this aggregator will damage the internal independence of features. On the other hand, the Connection Aggregator preserves complete feature information..

Summation: This aggregator is a traditional aggregator for feature aggregation. A summation aggregator is an operation of adding two feature matrices by adding the corresponding elements together. The calculation equation is shown as follows:

$$\mathbf{H_{agg}} = \mathbf{H^1_d} + \mathbf{H^1_f} \tag{9}$$

The dimension of the final result is the same as that of Hadamard. Therefore, the computational cost will be lower than the connection method for the subsequent training. However, the summation aggregation will also lead to the loss of information. To give an example, if $\mathbf{H^1_d}[\mathbf{i},\mathbf{j}] \gg \mathbf{H^1_f}[\mathbf{i},\mathbf{j}]$, then the elements in $\mathbf{H^1_f}$ will barely have an impact on the final feature value.

The aggregated feature representation $\mathbf{H_{agg}}$ is then used for classification. We map the representation to the predicted class label $\hat{Y}$ as follows:

$$\hat{Y} = \sigma(\mathbf{H_{agg}}\mathbf{W_o} + \mathbf{b_o})\mathbf{W_y} + \mathbf{b_y} \tag{10}$$

Here, $\sigma(\cdot)$ represents the activation function, $\mathbf{H_{agg}}$ is the feature vector obtained through the aggregation method, $\mathbf{W_o}$ and $\mathbf{b_o}$ are the weight matrix and bias vector used for linear transformation, $\mathbf{W_y}$ is the weight matrix of the output layer, and $\mathbf{b_y}$ is the bias vector of the output layer. With this combined formula, we can directly map the input feature $\mathbf{H_{agg}}$ to the final predicted class label $\hat{Y}$.

The loss function is shown below:

$$\mathcal{L}(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\hat{Y_i}), Y_i) \tag{11}$$

Here, $\ell(f(\hat{Y_i}), Y_i)$ computes the cross-entropy loss between the predicted and the ground-truth labels, where $f(\hat{Y_i})$ represents the predicted class probabilities. This loss function measures the dissimilarity between the predicted class probabilities and the actual labels.

5 Experiments

5.1 Experiment Setup

Datasets

We evaluate our model on four commonly used datasets for entity classification: AIFB, MUTAG, BGS, and PPI. Meanwhile, for link prediction, we evaluate our model on three datasets commonly used: FB15k-237, WN18, and WN18RR.

Baselines

We evaluate our model on the task of entity classification, comparing it with baselines such as R-GCN [Schlichtkrull *et al.*, 2018], RDF2Vec [Ristoski and Paulheim, 2016], TransE [Bordes *et al.*, 2013], and Comp-GCN [Vashishth *et al.*, 2019]. And we evaluate our model on the task of link prediction, comparing it with baselines such as DistMult [Yang *et al.*, 2015], ComplEx [Trouillon *et al.*, 2016], RotatE [Sun *et al.*, 2018], DualE [Cao *et al.*, 2021], GIE [Cao *et al.*, 2022], TDN [Wang *et al.*, 2023b], MSHE [Jiang *et al.*, 2024], MR-SAGCN [Song *et al.*, 2024a], and TCRA [Guo *et al.*, 2024].

Evaluation Metrics

In entity classification, we evaluate performance using accuracy, macro-precision, macro-recall, and macro-F1 for multi-classification tasks, and accuracy, precision, recall, and F1 for binary classification tasks, with higher values indicating better performance. For link prediction, we use four metrics: Mean Reciprocal Rank (MRR), and hit rates at 1, 3, and 10 (Hits@1, Hits@3, Hits@10), where higher values also indicate better performance.

5.2 Performance Evaluation of Entity Classification

Datasets	Entities	M31	M32	M41	M42	M43
AIFB	8,285	15,508	21,004	157	2,010	4,687
MUTAG	23,644	21,181	169,839	576	7,857	12,278
BGS	333,845	433,834	1,479,289	1,125	16,957	27,540
PPI	56,944	749,255	3,159,482	4,248	28,956	50,214
FB15k	14,541	188,001	358,181	5,102	24,196	30,584
WN18	40,943	1,378,722	2,195,504	8,769	28,807	67,593

Table 1: Entity motif degrees of six types of datasets.

Table 1 shows the entity motif degrees, Table 2 and Table 3 present the evaluation results of LORE and other baselines on four datasets, with the best metrics bolded.

For multi-classification tasks, LORE outperforms other baselines on the AIFB and PPI datasets. In AIFB, accuracy exceeds other methods by 3%-9%, and in PPI, by 1%-7%. Macro metrics also show improvements of 2%-10%. These results demonstrate that LORE accurately classifies entity types, with micro-F1 (equal to accuracy) and macro-F1 showing high performance, regardless of the balance of the data distribution.

In AIFB, the average entity motif degrees (M31, M32, M41, M42, and M43) are 1.87, 2.54, 0.02, 0.24, and 0.57, respectively, leading to higher classification accuracy due to

AIFB				
Method	Acc	Macro-Pre	Macro-Rec	Macro-F1
RDF2Vec	0.8888	0.8494	0.9	0.8672
R-GCN	0.9583	0.9391	0.8750	0.8983
TransE	0.9167	0.8862	0.9375	0.9148
Comp-GCN	0.9444	0.9	0.9583	0.9147
LORE	**0.9722**	**0.9583**	**0.9722**	**0.9692**
PPI				
Method	Acc	Macro-Pre	Macro-Rec	Macro-F1
RDF2Vec	0.5752	0.5494	0.6020	0.5757
R-GCN	0.7059	0.7391	0.6822	0.7107
TransE	0.6891	0.7015	0.6716	0.6866
Comp-GCN	0.7356	0.7029	0.7595	0.7312
LORE	**0.7687**	**0.7583**	**0.7722**	**0.7653**

Table 2: Performance evaluation in multi-classification tasks.

entities with similar higher-order relations being classified together. Conversely, in PPI, the network's density and complex protein relations limit the effectiveness of larger motifs, with LORE achieving a maximum accuracy of 76.87%.

For binary classification, LORE ranks second on MUTAG and first on BGS. On MUTAG, sparse relations and overfitting hinder LORE's performance, though it still performs well compared to Comp-GCN. In BGS, motifs with higher entity degrees (M31, M32) lead to improved performance, as the dataset is more sensitive to these higher-order logical relations, yielding better results than the baselines.

MUTAG				
Method	Acc	Precision	Recall	F1
RDF2Vec	0.6720	0.5	0.6528	0.5660
R-GCN	0.7323	0.5714	0.8696	0.6897
TransE	0.7353	0.6	0.6522	0.6250
Comp-GCN	**0.8650**	**0.8429**	**0.8826**	**0.8625**
LORE	0.8088	0.8	0.8696	0.8333
BGS				
Method	Acc	Precision	Recall	F1
RDF2Vec	0.8724	0.8750	0.7	0.7778
R-GCN	0.8310	0.7778	0.7	0.7368
TransE	0.8667	0.8	0.8	0.8
Comp-GCN	0.9	0.8889	0.8	0.8421
LORE	**0.9333**	**1.00**	**0.8**	**0.8889**

Table 3: Performance evaluation in binary classification tasks.

5.3 Performance Evaluation of Link Prediction

Table 4 shows the performance evaluation of link prediction. Our model achieves state-of-the-art results for all metrics in the FB15k-237 dataset, and for the WN18 dataset, our model performs best on the metrics of MRR, Hit@1, and Hit@3.

As shown in Table 4, the values of LORE on all of the FB15k-237 datasets are excellent and higher than all baselines. The results show that our method performs well in the link prediction task on the FB15k-237 dataset. The motif-based performance of LORE is also analyzed in this dataset. The knowledge graph represented by the FB15k-237 dataset is very sparse, as shown in Table 1, the average entity motif degrees of M31, M32, M41, M42, and M43 are 12.93, 24.63, 0.35, 1.67, and 2.1, respectively. For 237 relations, all the baselines and LORE can not accurately predict the relation. However, compared with other baselines, LORE still achieves better results. In the WN18 dataset, LORE achieves the highest values among all methods, with an MRR of 0.955, a Hit@1 of 0.944, and a Hit@3 of 0.965. As the WN18 dataset has fewer types of pairwise relationships, there are fewer types of higher-order logical relationships. However, the number of higher-order logical relationships is enormous. As shown in Table 1, the average entity motif degrees are

FB15K-237				
Model	MRR	Hits@10	Hits@3	Hits@1
R-GCN	0.248	0.417	0.264	0.151
TransE	0.294	0.465	-	-
DistMult	0.191	0.376	0.258	0.153
ComplEx	0.201	0.388	0.213	0.112
Comp-GCN	0.355	0.535	0.390	0.264
RotatE	0.338	0.533	0.375	0.241
DualE$_1$	0.326	0.512	0.357	0.235
DualE$_2$	0.359	0.552	0.391	0.264
GIE	0.364	0.553	0.401	0.270
TDN	0.350	0.546	0.395	0.263
MSHE	0.356	0.544	0.392	0.264
MR-SAGCN	0.368	0.550	0.403	0.276
TCRA	0.367	0.554	0.403	0.275
LORE	**0.386**	**0.567**	**0.412**	**0.308**
WN18				
Model	MRR	Hits@10	Hits@3	Hits@1
R-GCN	0.819	0.938	0.929	0.697
TransE	0.495	0.943	0.888	0.113
DistMult	0.813	0.943	0.921	0.701
ComplEx	0.941	0.947	0.936	0.936
Comp-GCN	0.930	**0.973**	0.931	0.732
RotatE	0.949	0.959	0.952	0.942
DualE$_1$	0.948	0.957	0.952	0.940
DualE$_2$	0.949	0.958	0.953	0.943
MSHE	0.948	0.957	0.951	0.943
LORE	**0.955**	0.948	**0.965**	**0.944**
WN18RR				
Model	MRR	Hits@10	Hits@3	Hits@1
R-GCN	0.358	0.430	0.388	0.312
TransE	0.232	0.533	0.409	0.022
DistMult	0.322	0.469	0.375	0.241
ComplEx	0.395	0.474	0.425	0.345
Comp-GCN	0.466	0.525	0.476	0.435
RotatE	0.474	0.570	0.492	0.426
DualE$_1$	0.475	0.542	0.491	0.440
DualE$_2$	0.486	0.555	0.502	0.449
GIE	0.494	**0.582**	**0.511**	0.448
TDN	0.481	0.481	0.502	0.439
MSHE	0.461	0.530	0.473	0.429
MR-SAGCN	0.489	0.563	0.505	0.450
TCRA	**0.496**	0.574	**0.511**	**0.457**
LORE	0.363	0.432	0.387	0.321

Table 4: Performance evaluation of the link prediction task on FB15K-237, WN18, and WN18RR datasets.

33.67, 53.62, 0.21, 0.7, and 1.65, respectively, so our method can accurately represent higher-order logical features in the representation process. On the WN18RR dataset, however, the performance of LORE is lower than that of other baselines. Considering that, compared with the WN18 dataset, the training set of WN18RR has no inverse triples, this result could partly prove that LORE utilizes inverse relations to achieve better performance.

6 Ablation Study

6.1 The Effectiveness of Formulating Higher-order Logical Relations with Motifs

LORE formulates higher-order logical relations using motifs. To assess their impact on the embedding results, we design four variants: **LORE-ID**, which uses the identity matrix as relational features; **LORE-M3**, which uses all three-order motifs; **LORE-M4**, which uses all four-order motifs (excluding four-node motifs with 3 edges); and **LORE-ALL**, which incorporates all motifs except the four-node motifs with 3 edges.

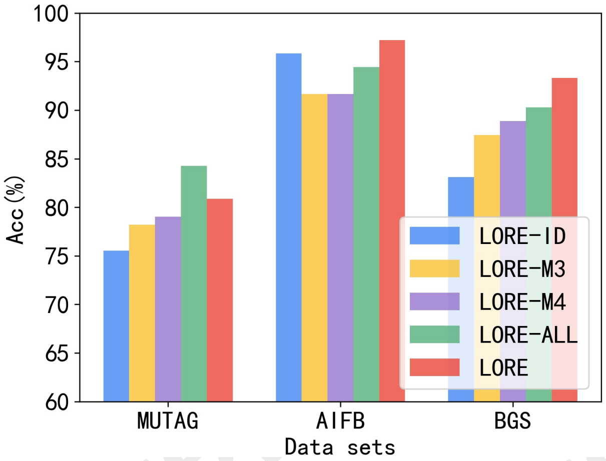


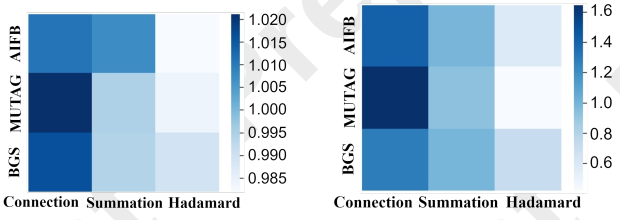
Figure 5: Accuracy of LORE and its variants.

Figure 5 shows the accuracy of LORE and its variants on three datasets. The results indicate that LORE achieves the second-highest accuracy on the MUTAG dataset, slightly below the LORE-ALL method, but outperforms all variants on the AIFB and BGS datasets.

We exclude four-node motifs with three edges and compare the accuracy of LORE-ID with the other variants to assess the effectiveness of higher-order logical relations. In MUTAG, LORE-ALL outperforms LORE-ID by 17%, while LORE surpasses LORE-ID by 18% in BGS. However, in AIFB, only LORE achieves higher accuracy than LORE-ID. These results show that formulating higher-order relations with motifs is beneficial, but improper motif selection may hinder representation learning.

On the AIFB and BGS datasets, LORE outperforms other variants using different motif selection strategies. In MUTAG, LORE's accuracy is lower than LORE-ALL. Despite LORE-ALL considering all motifs, its performance is worse than LORE on AIFB and BGS, possibly due to model overfitting or the sensitivity of these datasets to motif selection. This suggests that for different networks, motif selection is crucial for representation learning and can vary based on the task and dataset.

6.2 The Effectiveness of Aggregators



(a) Accuracy

(b) Time costs

Figure 6: Accuracy and time costs of three aggregators of LORE. (a) Accuracy; (b) Time costs.

LORE employs three aggregation methods: Connection, Hadamard, and Summation, referred to as LORE-C, LORE-H, and LORE-S, respectively. Figure 6 compares these methods after normalizing the calculated values. Figure 6a illustrates their accuracy, while Figure 6b depicts their time costs. Deeper colors indicate higher accuracy and greater time consumption. The Connection method achieves the highest accuracy across all datasets but incurs the highest time cost. The Summation method strikes a balance, with accuracy surpassing Hadamard but falling short of Connection, and a moderate time cost between the two. Hadamard offers the lowest time cost but also the lowest accuracy.

In summary, the Connection method best preserves structural and attribute features at the expense of higher time costs. The Hadamard method sacrifices more features but is computationally efficient, while the Summation method provides a middle ground. These aggregation methods can be selected based on specific application needs.

7 Conclusion

Existing knowledge graph representation learning methods cannot represent higher-order logical relations and entity attributes at the same time, leading to much representation loss. To solve this problem, we propose the LORE method, which uses motifs to formulate higher-order logical relations. In LORE, both the attribute features and relational features can be well represented. To aggregate these features, three feature aggregation methods are provided. Then, our method is evaluated on entity classification and link prediction tasks. In the entity classification task, our method is 2%–10% higher than baselines on accuracy. Also, our method is slightly higher than the baselines on MRR in the link prediction task. The results show that representation learning enhanced with higher-order logical relations formulation yields better performance than pure representation learning methods.

Acknowledgments

This study was supported by Hainan Provincial Natural Science Foundation of China (Grant No. 825CXTD608) and the grants (No. 62462022 and No. KYQD(ZR)-21079).

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[Cao *et al.*, 2021] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dual quaternion knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6894–6902, 2021.

[Cao *et al.*, 2022] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Geometry interaction knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5521–5529, 2022.

[Ding *et al.*, 2024] Wentao Ding, Jinmao Li, Liangchuan Luo, and Yuzhong Qu. Enhancing complex question answering over knowledge graphs through evidence pattern retrieval. In *Proceedings of the ACM on Web Conference 2024*, pages 2106–2115, 2024.

[Donnat *et al.*, 2018] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1320–1329, 2018.

[Guo *et al.*, 2024] Jingtao Guo, Chunxia Zhang, Lingxi Li, Xiaojun Xue, and Zhendong Niu. A unified joint approach with topological context learning and rule augmentation for knowledge graph completion. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13686–13696, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Jiang *et al.*, 2024] Dan Jiang, Ronggui Wang, Lixia Xue, and Juan Yang. Multisource hierarchical neural network for knowledge graph embedding. *Expert Systems with Applications*, 237:121446, 2024.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li *et al.*, 2024] Yicong Li, Xiangguo Sun, Hongxu Chen, Sixiao Zhang, Yu Yang, and Guandong Xu. Attention is not the only choice: counterfactual reasoning for path-based explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Liao *et al.*, 2021] Zihan Liao, Wenxin Liang, Han Liu, Jie Mu, and Xianchao Zhang. Self-supervised graph representation learning with variational inference. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 116–127. Springer, 2021.

[Liu *et al.*, 2024] Yuhan Liu, Zelin Cao, Xing Gao, Ji Zhang, and Rui Yan. Bridging the space gap: Unifying geometry knowledge graph embedding with optimal transport. In *Proceedings of the ACM on Web Conference 2024*, pages 2128–2137, 2024.

[Niu *et al.*, 2021] Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 213–222, 2021.

[Qiu *et al.*, 2020] Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 474–482, 2020.

[Ranganathan and Barbosa, 2022] Varun Ranganathan and Denilson Barbosa. Hoplop: multi-hop link prediction over knowledge graph embeddings. *World Wide Web*, 25(2):1037–1065, 2022.

[Ren *et al.*, 2021] Jing Ren, Feng Xia, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong. Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(3):332–350, 2021.

[Ristoski and Paulheim, 2016] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*, ISWC '16, pages 498–514. Springer, 2016.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, ESWC '18, pages 593–607. Springer, 2018.

[Shan *et al.*, 2023] Minghui Shan, Yixiao Ma, Shulan Ruan, Zhi Cao, Shiwei Tong, Qi Liu, Yu Su, and Shijin Wang. Paperlm: A pre-trained model for hierarchical examination paper representation learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2178–2187, 2023.

[Shi *et al.*, 2017] Jun Shi, Huan Gao, Guilin Qi, and Zhangquan Zhou. Knowledge graph embedding with triple context. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2299–2302, 2017.

[Shokrzadeh *et al.*, 2024] Zeinab Shokrzadeh, Mohammad-Reza Feizi-Derakhshi, Mohammad-Ali Balafar, and Jamshid Bagherzadeh Mohasefi. Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding. *Ain Shams Engineering Journal*, 15(1):102263, 2024.

[Song *et al.*, 2024a] Jiawei Song, Zongtao Duan, Jianrong Cao, and Yun Lin. Multi-relational semantic awareness for knowledge graph completion. In *2024 9th International Conference on Computer and Communication Systems (ICCCS)*, pages 107–113, 2024.

[Song *et al.*, 2024b] Xiangxiang Song, Guang Ling, Wenhui Tu, and Yu Chen. Knowledge-guided heterogeneous graph convolutional network for aspect-based sentiment analysis. *Electronics*, 13(3):517, 2024.

[Sun *et al.*, 2018] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, ICLR '18. PMLR, 2018.

[Sun *et al.*, 2022] Ke Sun, Shuo Yu, Ciyuan Peng, Yueru Wang, Osama Alfarraj, Amr Tolba, and Feng Xia. Relational structure-aware knowledge graph representation in complex space. *Mathematics*, 10(11):1930, 2022.

[Taunk *et al.*, 2023] Dhaval Taunk, Lakshya Khanna, Siri Venkata Pavan Kumar Kandru, Vasudeva Varma, Charu Sharma, and Makarand Tapaswi. Grapeqa: Graph augmentation and pruning to enhance question-answering. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1138–1144, 2023.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, ICML '16, pages 2071–2080. PMLR, 2016.

[Tu *et al.*, 2023] Huiling Tu, Shuo Yu, Vidya Saikrishna, Feng Xia, and Karin Verspoor. Deep outdated fact detection in knowledge graphs. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1443–1452, Shanghai, China, 2023. IEEE.

[Vashishth *et al.*, 2019] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, ICLR '19. PMLR, 2019.

[Wang *et al.*, 2023a] Guiling Wang, Jian Yu, Mo Nguyen, Yuqi Zhang, Sira Yongchareon, and Yanbo Han. Motif-based graph attentional neural network for web service recommendation. *Knowledge-Based Systems*, 269:110512, 2023.

[Wang *et al.*, 2023b] Jiapu Wang, Boyue Wang, Junbin Gao, Xiaoyan Li, Yongli Hu, and Baocai Yin. Tdn: Triplet distributor network for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Wang *et al.*, 2023c] Suixue Wang, Xiangjun Hu, and Qingchen Zhang. Hc-mae: Hierarchical cross-attention masked autoencoder integrating histopathological images and multi-omics for cancer survival prediction. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 642–647, 2023.

[Wang *et al.*, 2024] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214, 2024.

[Xu *et al.*, 2020] Da Xu, Chuanwei Ruan, Jason Cho, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Knowledge-aware complementary product representation learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 681–689, 2020.

[Yan *et al.*, 2024] Pengwei Yan, Kaisong Song, Zhuoren Jiang, Yangyang Kang, Tianqianjin Lin, Changlong Sun, and Xiaozhong Liu. Empowering dual-level graph self-supervised pretraining with motif discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9223–9231, 2024.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, ICLR '15. PMLR, 2015.

[Yu *et al.*, 2020] Shuo Yu, Feng Xia, Jin Xu, Zhikui Chen, and Ivan Lee. Offer: A motif dimensional framework for network representation learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3349–3352, 2020.

[Zhao *et al.*, 2023] Na Zhao, Zhen Long, Jian Wang, and Zhi-Dan Zhao. Agre: A knowledge graph recommendation algorithm based on multiple paths embeddings rnn encoder. *Knowledge-Based Systems*, 259:110078, 2023.

[Zheng *et al.*, 2018] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Zhong *et al.*, 2023] Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, Hua Jin, and Dacheng Tao. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Zong *et al.*, 2024] Linlin Zong, Zhenrong Xie, Chi Ma, Xinyue Liu, Xianchao Zhang, and Bo Xu. Renn: A rule embedding enhanced neural network framework for temporal knowledge graph completion. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13919–13928, 2024.