Dual-Hypergraph Embedding for Hyper-Relational Knowledge Graphs

Anonymous ACL submission

Abstract

Hyper-relational knowledge graphs (HKGs) have been shown to effectively represent complex data relation. However, many HKG representation learning methods ignore the one-to-many relation between attributes and values in hyper-relational facts (H-Fact). Moreover, the information extraction from H-Facts is insufficient. To solve these issues, we propose a **D**ual-**H**ypergraph **E**mbedding for HKGs (DHE). Specifically, we first construct an intra-hypergraph and an inter-hypergraph by considering one-to-many relation in H-Facts, and then design a hypergraph Transformer encoder to comprehensively conduct H-Fact representation learning. Experimental results on real datasets show that our DHE outperforms all SOTA methods in link prediction tasks on HKGs. Source code can be found at the url: `https://anonymous.4open.science/r/DHE`.

1 Introduction

Traditional triple-based Knowledge Graph (KG) is a flexible way to explore data relation for link prediction, by using a triple (s, r, o) to represent the binary relation r between subjects s and objects o (Li et al., 2023). However, it often fails to accurately represent complex data relation, especially for dealing with hyper-relational facts (H-Facts), consisting of a main triple (s, r, o) and qualifiers, which provide detailed information to describe the main triple by attributes a and values v. Recently, Hyper-relational Knowledge Graphs (HKGs) have been widely designed to conduct link prediction in H-Facts (Rosso et al., 2020).

Based on the ways of the H-Fact definition, previous HKG methods can be grouped into two categories, *i.e.,* transferred HKG methods and combinated HKG methods. Transferred HKG methods transfer the H-Fact to either an n-tuple (*e.g.,* m-TransH (Wen et al., 2016) and RAE (Zhang et al., 2018)) or a set of key-value pairs (*e.g.,* NaLP (Guan et al., 2019) and RAM (Liu et al., 2021)) to learn latent representations of entities and relations of the H-Fact. However, they equally treat every qualifier and thus easily destroying the original structural information of H-Facts (Xiong et al., 2023). Combinated HKG methods combine the main triple (*i.e.,* (s, r, o)) with qualifiers (*i.e.,* $\{\{a_i, v_i\}_{i=1}^m\}$) to obtain a tuple, *i.e.,* $\{(s, r, o), \{a_i, v_i\}_{i=1}^m\}$, for directly learning representation for the H-Fact. For instance, GRAN (Wang et al., 2021) represents H-Facts as a heterogeneous graph and HAHE (Luo et al., 2023) integrates global and local information among H-Facts.

Although previous transferred and combinated HKG methods have been widely applied, they still have limitations to be addressed. First, they ignore one-to-many relation between attributes and values in H-Facts, so they are difficult to capture crucial information in H-Facts. As shown in Figure 2a, attribute "*nominee*" has four values, previous methods (*e.g.,* GRAN and HAHE) describe the H-Fact by four attribute-value pairs, leading to the issue of attribute redundancy, *i.e.,* three more "*nominee*". Second, it is difficult for previous methods to accurately capture the information in H-Facts based on main triples and qualifiers. For example, previous methods (*e.g.,* GRAN) can not use both the main triple and the qualifier "{*point in time: 2001*}" in Figure 2a to infer the time information of the H-Fact. The reason is that pairwise relation in GRAN is not enough to characterize such complex relation in the H-Fact of Figure 2a (Zhou et al., 2023).

To address the above issues, in this paper, we propose a new combinated HKG method, namely **D**ual-**H**ypergraph **E**mbedding for HKGs, namely DHE, shown in Figure 1. Specifically, based on the observation that many H-Facts in the HKGs have one-to-many relation between the attributes and its values, we define one-to-many H-Facts to illustrate such one-to-many relation, exploring the first

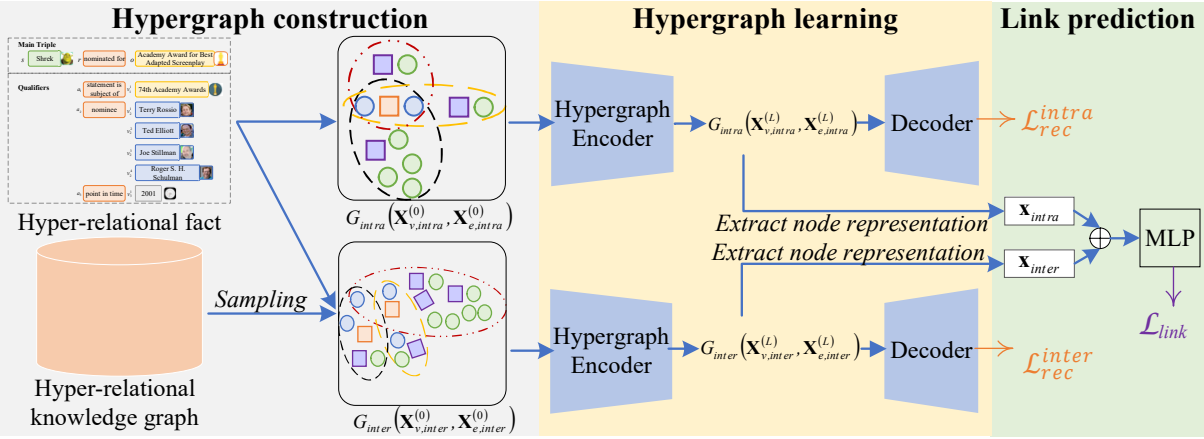


Figure 1: The flowchart of the proposed DHE, consisting of three components, *i.e.,* hypergraph construction, hypergraph learning and link prediction. Specifically, hypergraph construction constructs an intra-hypergraph G_{intra} and an inter-hypergraph G_{inter}, considering the one-to-many relation in H-Facts. Hypergraph learning investigates a new Transformer encoder to comprehensively learn representations of H-Facts for two hypergraphs. Furthermore, two kinds of representations are fused for conducting link prediction.

issue in previous methods. Based on the new definition, we construct an inter-hypergraph and an intra-hypergrpah, which are further extracted two kinds of representations by a proposed hypergraph Transformer encoder. As a result, the inter-hypergraph considers all elements in H-facts and node diversity to explore the relation among H-Facts, and the intra-hypergraph explores the high-order information within individual H-Fact. Hence, both of them explore the second issue in previous methods by comprehensively extracting the information in H-Facts. The main contributions of our method are summarized as follows.

- Based on the real scenarios that many H-Facts have one-to-many relation between attributes and values, we propose a new definition of H-Facts to capture such complex relation for link prediction on HKGs.
- We consider one-to-many relation to investigate two hypergraphs for comprehensively capturing the relation among H-Facts (by the inter-hypergraph) and the high-order information within individual H-Fact (by the intra-hypergraph). Experimental results on three datasets demonstrate the effectiveness of our proposed method for link prediction tasks.

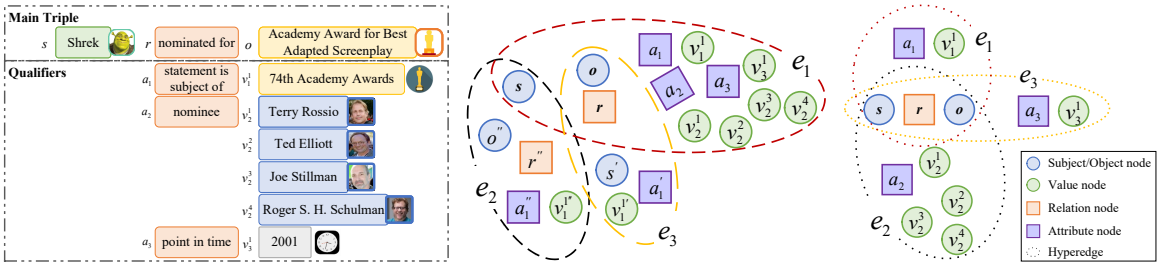
2 Method

Previous HKG methods for link prediction usually include four steps, *i.e.,* H-Fact definition, heterogeneous graph construction, representation learning and link prediction. Specifically, given the definition of an H-Fact as $\mathcal{F} = \{(s, r, o), \{a_i, v_i\}_{i=1}^m\}$, a heterogeneous graph is constructed to characterize and represent H-Facts. After that, deep methods (*e.g.,* graph neural networks and Transformer (Vaswani et al., 2017)) are used to learn the representation of the H-Fact for capturing its structural and semantic information. Finally, link prediction is conducted on the learned representations.

2.1 One-to-many H-Facts

Previous HKG methods (Wang et al., 2021; Luo et al., 2023) define the H-Fact as $\mathcal{F} = \{\mathcal{T}, \mathcal{Q}\}$, where $\mathcal{T} = \{(s, r, o)|s, o \in E, r \in R\}$ is a main triple, $\mathcal{Q} = \{\{a_i, v_i\}_{i=1}^m | v_i \in E, a_i \in R\}$ is the qualifiers, m is the number of qualifiers, E is the entity sets, and R is the relation sets. Obviously, qualifiers $\{a_i, v_i\}_{i=1}^m$ describe one-to-one relation between the attribute a_i and the value v_i. However, many H-Facts in the HKGs have one-to-many relation between the attributes and its values. For example, the attribute "*nominee*" in the qualifier of the H-Fact is with four values, *i.e.,* "*Terry Rossio, Ted Elliott, Joe Stillman, Roger S.H. Schulman*" in Figure 2a, the previous definition represents it by four one-to-one qualifiers, *i.e.,* "{*nominee: Terry Rossio*}, {*nominee: Ted Elliott*}, {*nominee: Joe Stillman*}, and {*nominee: Roger S.H. Schulman*}". As a result, the attribute redundancy (*i.e.,* three more "*nominee*") may affect representation learning of H-Facts. To address the above issue, we first define the one-to-many relation as follows.

Definition 1 (One-to-many H-Facts) *A one-to-many H-Fact is defined as* $\mathcal{F} = \{\mathcal{T}, \mathcal{Q}\}$*, where* $\mathcal{T} = \{(s, r, o)|s, o \in E, r \in R\}$ *is a main triple,* $\mathcal{Q} = \{\{a_i, \{v_i^j\}_{j=1}^{n_i}\}_{i=1}^m | v_i^j \in E, a_i \in R\}$ *is a*



(a) An example of H-Fact (b) The proposed inter-hypergraph (c) The proposed intra-hypergraph

Figure 2: An illustration of our proposed hypergraph construction on an H-Fact: (a) Different definitions on the H-Fact, *i.e.,* (s, r, o) for triple-based KG methods, either $r(s, o, v_1^1, v_2^1, v_2^2, v_2^3, v_2^4, v_3^1)$ or $\{(r : s), (r : o), (a_1 : v_1^1), (a_2 : v_2^1), (a_2 : v_2^2), (a_2 : v_2^3), (a_2 : v_2^4), (a_3 : v_3^1)\}$ for transferred HKG methods, $\{(s, r, o), \{a_1, v_1^1\}, \{a_2, v_2^j\}_{j=1}^4, \{a_3, v_3^1\}\}$ for combinated HKG methods, and $\{(s, r, o), \{a_1, v_1^1\}, \{a_2, v_2^1, v_2^2, v_2^3, v_2^4\}, \{a_3, v_3^1\}\}$ for our method. (b) The proposed inter-hypergraph based on the H-Fact listed in Figure 2a, which is represented by Hyperedge e_1 connecting all its elements. Hyperedges (*i.e.,* e_2 and e_3) are two H-Facts adjacent to the H-Fact. (c) The proposed intra-hypergraph based on the H-Fact listed in Figure 2a, where three hyperedges (*i.e.,* e_1, e_2 and e_3) indicate aspects of individual qualifiers of the H-Fact.

set of qualifiers, and m and n_i, respectively, is the number of qualifiers and the values for the i-th qualifier. Note that, the number of values of different qualifiers are different.

In Definition 1, $\mathcal{Q}$ differs from previous methods. Specifically, $\mathcal{Q}$ in previous methods is one-to-one relation. In contrast, in our Definition 1, each qualifier consists of an attribute a_i and a number of values $\{v_i^j\}_{j=1}^{n_i}$, so their relation is one-to-many. Based on our Definition 1, the attribute "*nominee*" in Figure 2a is a one-to-many qualifier, *i.e.,* "{*nominee: Terry Rossio, Ted Elliott, Joe Stillman, Roger S.H. Schulman*}", reducing the issue of attribute redundancy. Moreover, previous definition is one of special issues of our Definition 1, *i.e.,* $n_i = 1$ for the i-th qualifier. In particular, the triple-based KG is actually a triple with $m = 0$.

2.2 Hypergraph construction

Given Definition 1, we first follow (Luo et al., 2023; Chen et al., 2023) to construct a hypergraph, where the elements of H-Facts form the node set of the hypergraph and the information of the H-Fact is characterized by the hyperedge, and then conduct representation learning for H-Facts based on the constructed hypergraph.

Inspired by previous works (Luo et al., 2023; Hu et al., 2023), representation learning for H-Facts on HKGs should explore all kinds of relation among H-Facts for comprehensively understanding of the H-Facts. To achieve this, we construct two hypergraphs based on Definition 1, *i.e.,* an inter-hypergraph exploring the relation among H-Facts and an intra-hypergraph exploring the relation among individual H-Fact.

2.2.1 Inter-hypergraph construction

Previous methods (*i.e.,* HAHE (Luo et al., 2023)) focus on capturing the relation among H-Facts to 1) construct an inter-hypergraph based on a part of elements of H-Facts, including subjects, objects and values $\mathcal{V} = \{s, o, \{v_1^j\}_{j=1}^{n_1}, \cdots, \{v_m^j\}_{j=1}^{n_m}\}$ by ignoring the relation and attributes; 2) equivalently regard all kinds of nodes by ignoring their node diversity; and 3) consider the relation between every H-Fact and any other H-Facts to result in computation inefficiency.

First, both relations and attributes are useful for representation learning. For example, in Figure 2a, the attribute "*nominee*" constrains the role of the value "*Terry Rossio*" as a nominee. Second, the elements of every H-Fact in HKGs include subjects, objects, relations, attributes and values. Obviously, they have different contributions for representation learning. For example, in the H-Fact of Figure 2a, relation "*nominated for*" indicates the specific relation between subject "*Shrek*" and object "*Academy Award for Best Adapted Screenplay*", which is the core information of the H-Fact. However, attribute "*point in time*" and value "*2001*" are the supplementary information of the H-Fact. Third, exploring the relation between every H-Fact and all other H-Facts makes computation inefficiency as well as the model complex.

In this paper, we construct an inter-hypergraph to address the above issues by the following steps. Specifically, we observe that H-Facts in HKGs are interconnected to each other. Moreover, such interconnectedness is crucial information for representa-

tion learning of H-Facts on HKGs (Lu et al., 2023). For example, assuming that the H-Fact in Figure 2a is the adjacency of this H-Fact, *i.e.,* "{*(Shrek, voice actor, Mike Myers*), {*character role: Shrek*}}", we easily know that this H-Fact provides complement information to the H-Fact in Figure 2a, *i.e.,* the information of voice actor of the Shrek character. Hence, the representation learning of every H-Fact can be strenghthened by the information of the adjacent H-Fact (Hu et al., 2023). However, to the best of our knowledge, no literature are designed to describe such interconnectedness. In this paper, we investigate Definition 2 to describe the adjacent relation between two H-Facts:

Definition 2 (H-Fact adjacency) *Given two H-Facts* $\mathcal{F}_1 = \{\mathcal{T}_1, \mathcal{Q}_1\}$ *and* $\mathcal{F}_2 = \{\mathcal{T}_2, \mathcal{Q}_2\}$*, where* $\mathcal{T}_1 = \{s_1, r_1, o_1\}$ *and* $\mathcal{T}_2 = \{s_2, r_2, o_2\}$ *are the main triples, and* $\mathcal{Q}_1$ *and* $\mathcal{Q}_2$ *are the qualifiers, if* $\mathcal{T}_1 \cap \mathcal{T}_2 \neq \emptyset$*,* $\mathcal{F}_1$ *is adjacent to* $\mathcal{F}_2$*.*

Based on Definition 2, we construct an inter-hypergraph $\mathcal{G}_{inter} = (\mathcal{V}_{inter}, \mathcal{E}_{inter}, \mathbf{H}_{inter})$ for the H-Fact by the following steps.

(1) We randomly sample k neighbours for every H-Fact to form its H-Fact set $F = \{\mathcal{F}, \mathcal{F}_1, \cdots, \mathcal{F}_k\}$, whose corresponding hyperedge set is $\mathcal{E}_{inter}$=$\{e, e_1, \cdots, e_k\}$. As a result, the number of hyperedges is $(k+1)$. Hence, different from previous methods that capture the global relation among H-Facts, our method captures the local relation, *i.e.,* the relation among the H-Fact and its k adjacent H-Facts. As a result, our method achieves computation efficiency and model simplicity , *i.e.,* k=10 is enough rather than all neighbours (Verified in Section 3.2).

(2) The node set $\mathcal{V}_{inter}$ consists of all elements of the H-Fact set F. Every node is classified into one of four categories, *i.e.,* {*subject/object node, relation node, attribute node, and value node*}. Since the same entity (*e.g.,* "*Shrek*" in Figure 2a) may be a subject or an object in different triples, so it is reasonable to group the subject and the object in the same category.

(3) The incidence matrix $\mathbf{H}_{inter} \in \mathbb{R}^{|\mathcal{V}_{inter}| \times (k+1)}$ indicates the topology of the inter-hypergraph. Its element is obtained by:

$$\begin{cases} h_{ij} = 1, \text{ if } v_i \in \mathcal{F}_j \\ h_{ij} = 0, \text{ if } v_i \notin \mathcal{F}_j \end{cases} \tag{1}$$

Figure 2b is the illustration of the proposed inter-hypergraph for the H-Fact in Figure 2a based on the above steps. Specifically, the H-Fact in Figure 2a has two adjacency, *i.e.,* e_2 and e_3. The node set contains all elements of the H-Fact set F, *i.e.,* $\mathcal{V} = \{s, r, o, a_1, \cdots, a_m, \{v_1^j\}_{j=1}^{n_1}, \cdots, \{v_m^j\}_{j=1}^{n_m}\}$.

2.2.2 Intra-hypergraph construction

Previous work construct the normal graphs (Wang et al., 2021) or semantic sequences (Wei et al., 2023) to consider the relation within individual H-Fact, by ignoring high-order information within individual H-Fact. For example, in Figure 2a, combining the qualifier "{*point in time*: *2001*}" with the main triple indicates the temporal information of "*Shrek nominated for Academy Award for Best Adapted Screenplay*". However, the existing methods only capture part information among this qualifier and the main triple, *e.g.,* the value "*2001*" and the attribute "*point in time*", because they construct the normal graph to connect two nodes, *i.e.,* the value and the attribute. Obviously, the former information is useful than the last information for representation learning.

In this paper, we construct an intra-hypergraph $\mathcal{G}_{intra}$=$(\mathcal{V}_{intra}, \mathcal{E}_{intra}, \mathbf{H}_{intra})$ (illustrated in Figure 2c) to capture high-order information within the H-Fact $\mathcal{F}$=$\{(s, r, o), \{a_i, \{v_i^j\}_{j=1}^{n_i}\}_{i=1}^{m}\}$ based on Definition 1.

(1)The node set is denoted as $\mathcal{V}_{intra} = \{s, r, o, a_1, \cdots, a_m, \{v_1^j\}_{j=1}^{n_1}, \cdots, \{v_m^j\}_{j=1}^{n_m}\}$, where every node $v_i \in \mathcal{V}_{intra}$ corresponds to an element of H-Fact $\mathcal{F}$. The node set $\mathcal{V}_{intra}$ consists of four types, *i.e.,* {*subject/object node, relation node, attribute node, and value node*}.

(2)The hyperedge set $\mathcal{E}_{intra} = \{e_1, \cdots, e_m\}$ consists of the main triple (s, r, o) and one qualifier $\{a_i, \{v_i^j\}_{j=1}^{n_i}\}$, *i.e.,* $e_1 = \{s, r, o, a_1, \{v_1^j\}_{j=1}^{n_1}\}$. Each hyperedge $e_i \in \mathcal{E}_{intra}$ represents specific information of the H-Fact $\mathcal{F}$.

(3)The incidence matrix $\mathbf{H}_{intra} \in \mathbb{R}^{|\mathcal{V}_{intra}| \times m}$ indicates the topology of the intra-hypergraph and its element is obtained by:

$$\begin{cases} h_{ij} = 1, \text{ if } v_i \in e_j \\ h_{ij} = 0, \text{ if } v_i \notin e_j \end{cases} \tag{2}$$

Different from previous works (Wei et al., 2023), our hypergraph construction shows the following advantages. First, the hyperedge treats attributes and values in the qualifier as a group, which contains a varied number of values to explore the one-to-many relationship. Second, the intra-hypergraph focuses on high-order information within individual H-Fact and the intra-hypergraph focuses on adjacency information among different H-Facts,

so these two hypergraphs provide complementary information to each other.

2.3 Hypergraph learning

Given two hypergraphs, previous methods (Luo et al., 2023) usually design two separated encoders in a pipeline to learn representation for H-Facts. This easily leads to lose the information for subsequent representation learning. To address this issue, we investigate a new hypergraph Transformer encoder to simultaneously extract inter-representation and intra-representation, respectively, from the inter-hypergraph and the intra-hypergraph. Specifically, each hypergraph Transformer has one encoder and one decoder, where the encoder includes initialisation, node biases, multi-head hypergraph attention, Layer Normalisation (LN) and Feed-Forward Network (FFN). The decoder includes an inner-product operation. Since two hypergraph Transformers have the same encoder and decoder, for simplicity, we use the symbol $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{H})$ to denote either the inter-hypergraph and the intra-hypergraph in this section.

2.3.1 Hypergraph encoder

Initialisation. Given the input (*i.e.,* $\mathcal{G}$=$(\mathcal{V}, \mathcal{E}, \mathbf{H})$) from either hypergraph, we first employ the input embedding layer (*i.e.,* word embedding) to map all elements in the node set $\mathcal{V}$ of the hypergraph into continuous vector representations $\mathbf{X}_v^{(0)} \in \mathbb{R}^{|\mathcal{V}| \times d_v}$, where $|\mathcal{V}|$ is the number of nodes of the hypergraph, and d_v is the dimension of the nodes. We then adopt the hyperedge initialisation method in Zhou et al. (2023) to obtain hyperedge representation matrix $\mathbf{X}_e^{(0)} \in \mathbb{R}^{|\mathcal{E}| \times d_e}$, where $|\mathcal{E}|$ is the number of hyperedges of the hypergraph, and d_e is the dimension of the hyperedges. We further feed the initialised $\mathbf{X}_v^{(0)}$ and $\mathbf{X}_e^{(0)}$, as well as the incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ into the new Transformer encoder with L layers to update the representation for both nodes and hyperedges.

Node biases. In the hypergraph construction, the elements of H-Fact have four types of nodes, *i.e., subject/object node, relation node, attribute node, and value node*, so we design a mapping function $\psi(\cdot)$ to assign every node with node biases by:

$$\psi(v_i) = \begin{cases} (o_1^Q, o_1^K), & \text{if } v_i \text{ is subject/object node} \\ (o_2^Q, o_2^K), & \text{if } v_i \text{ is relation node} \\ (o_3^Q, o_3^K), & \text{if } v_i \text{ is attribute node} \\ (o_4^Q, o_4^K), & \text{if } v_i \text{ is value node} \end{cases} \tag{3}$$

where $o_u^Q \in \mathbb{R}^{d_z}$ and $o_u^K \in \mathbb{R}^{d_z}$ $(u = 1, 2, 3, 4)$ are the query and the key biases of node v_i, respectively, $d_z = \frac{d_v}{T}$ is the dimension of the heads in all layers, T is the number of the heads.

Specifically, we first employ the word embedding layer to obtain the representation vector for every type of the node v_i, *i.e.,* (o_u^Q, o_u^K), and then integrate them to calculate the attention coefficient. This representation vector is updated during the back-propagation process of the trained model.

Multi-head hypergraph attention. Given initialised $\mathbf{X}_v^{(0)}$ and $\mathbf{X}_e^{(0)}$, and node biases o_u^Q and o_u^K, we design the multi-head hypergraph attention with T heads to learn node and hyperedge representations for every H-Fact.

(1) Hyperedge representation. The hyperedge representation has been demonstrated to effectively capture the relation among H-Facts or the relation within individual H-Fact (Luo et al., 2023). However, previous work (*e.g.,* HAHE (Luo et al., 2023)) did not consider the node diversity for learning hyperedge representation, which may affect the effectiveness of the attention coefficient.

To address the above issue, we add node biases in Eq. (3) to calculate the attention coefficient $a_{e_i,v_j}^{(l)}$ in the l-th layer between the j-th node and the i-th hyperedges by:

$$a_{e_i,v_j}^{(l)} = \frac{(\mathbf{W}_e^Q e_i) \cdot (\mathbf{W}_v^K v_j + o_u^K)^\top}{\sqrt{d_z}} \tag{4}$$

where $\mathbf{W}_e^Q \in \mathbb{R}^{d_z \times d_e}$ and $\mathbf{W}_v^K \in \mathbb{R}^{d_z \times d_v}$ are the learnable weight, $e_i \in \mathbf{X}_e^{(l-1)}$ is the i-th hyperedge representation, $v_j \in \mathbf{X}_v^{(l-1)}$ is the j-th node representation, and o_u^K is designed as the key bias of the node v_j by Eq. (3).

Different types of nodes have different effects on the hyperedge representation, so we add o_u^K in Eq. (3) to learn the attention coefficient by Eq. (4), aiming at achieving the node diversity. We further aggregate the representations of all nodes in every hyperedge by

$$\hat{\mathbf{X}}_e^{(l,t)} = \text{Softmax}(\mathbf{A}_e^{(l)} \cdot (\mathbf{W}_v^V \cdot \mathbf{X}_v^{(l-1)}) \tag{5}$$

where $\mathbf{W}_v^V \in \mathbb{R}^{d_z \times d_v}$ is the learnable weight, $\mathbf{A}_e^{(l)} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{V}|}$ is the attention matrix of hyperedges and $a_{e_i,v_j}^{(l)}$ is the element of $\mathbf{A}_e^{(l)}$ in the i-th row and the j-th column. After that, we aggregate $\hat{\mathbf{X}}_e^{(l,t)}$ and $\mathbf{X}_e^{(l-1,t)}$ as the hyperedge representation matrix $\mathbf{X}_e^{(l,t)}$ of the t-th head and in the l-th layer:

$$\mathbf{X}_e^{(l,t)} = \text{MLP}(\text{Concat}(\hat{\mathbf{X}}_e^{(l,t)}, \mathbf{X}_e^{(l-1,t)})) \tag{6}$$

where $\mathrm{MLP}(\cdot)$ is multi-layer perceptron, $\mathrm{Concat}(\cdot)$ means the concatenation operation. Therefore, the hyperedge representation of the l-layer can be updated by the following formula:

$$\mathbf{X}_e^{(l)} = \sigma(\mathbf{W}_e^{(l)} \cdot (\mathrm{Concat}(\mathbf{X}_e^{(l,1)}, \cdots, \mathbf{X}_e^{(l,T)}))) \tag{7}$$

where $\mathbf{W}_e^{(l)}$ is the multi-head weight of the hyperedges, and $\sigma(\cdot)$ is the activation function GELU (Hendrycks and Gimpel, 2016) in this paper.

(2) Node representation. After obtaining the hyperedge representations $\mathbf{X}_e^{(l)}$ in the l-th layer, we aggregate the representation of the hyperedge into the node representations of the hypergraph to achieve information fusion. Similarly, we also add the node biases in Eq. (3) to calculate attention coefficients $a_{v_i,e_j}^{(l)}$ in the l-th layer between the j-th hypergraph and the i-th nodes by

$$a_{v_i,e_j}^{(l)} = \frac{(\mathbf{W}_v^Q v_i + o_u^Q) \cdot (\mathbf{W}_e^K e_j)^\top}{\sqrt{d_z}} \tag{8}$$

where $\mathbf{W}_v^Q \in \mathbb{R}^{d_z \times d_v}$ and $\mathbf{W}_e^K \in \mathbb{R}^{d_z \times d_e}$ are the learnable weight, $e_j \in \mathbf{X}_e^{(l)}$ is the j-th hyperedge representation in the l-th layer, $v_i \in \mathbf{X}_v^{(l-1)}$ is the i-th node representation, and o_u^Q is designed as the quary bias of the node v_i. We aggregate the representations of all hyperedges for each node by:

$$\mathbf{X}_v^{(l,t)} = \mathrm{Softmax}(\mathbf{A}_v^{(l)}) \cdot (\mathbf{W}_e^V \cdot \mathbf{X}_e^{(l)}) \tag{9}$$

where $\mathbf{W}_e^V \in \mathbb{R}^{d_z \times d_e}$ is the learnable weight, $\mathbf{A}_v^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ is the attention matrix of nodes and $a_{v_i,e_j}^{(l)}$ is the element of $\mathbf{A}_v^{(l)}$ in the i-th row and the j-th columen. After that, we concatenate T heads to obtain the node representation matrix:

$$\bar{\mathbf{X}}_v^{(l)} = \sigma(\mathbf{W}_v^{(l)} \cdot (\mathrm{Concat}(\mathbf{X}_v^{(l,1)}, \cdots, \mathbf{X}_v^{(l,T)}))) \tag{10}$$

where $\mathbf{W}_v^{(l)}$ is the multi-head weight of the nodes.

Layer normalisation and Feed-Forward network. Given the node representation matrix $\mathbf{X}_v^{(l-1)}$ in the $(l-1)$-th layer and the node representation matrix $\bar{\mathbf{X}}_v^{(l)}$ calculated by Eq. (10), similar to Transformer (Vaswani et al., 2017), we introduce the layer normalisation (LN) and the feed-forward network (FFN) , *i.e.,*

$$\tilde{\mathbf{X}}_v^{(l)} = \mathrm{LN}(\bar{\mathbf{X}}_v^{(l)} + \mathbf{X}_v^{(l-1)}) \tag{11}$$

$$\mathbf{X}_v^{(l)} = \mathrm{LN}(\mathrm{FFN}(\tilde{\mathbf{X}}_v^{(l)}) + \tilde{\mathbf{X}}_v^{(l)}) \tag{12}$$

After L layers, we obtain node representation $\mathbf{X}_v^{(L)}$ by Eq. (12) and hyperedge representation $\mathbf{X}_e^{(L)}$ by Eq. (7) for the hypergraph $\mathcal{G}$.

2.3.2 Hypergraph decoder

Given the node representation $\mathbf{X}_v^{(L)}$ and the hyperedge representation $\mathbf{X}_e^{(L)}$ for all H-Facts, we design a hypergraph decoder (*i.e.,* inner-product operation) to reconstruct the hypergraph topological structure $\mathbf{H}$ so that the decoder helps to learn useful representations for link prediction tasks (Fan et al., 2021). Specifically, denoting $\hat{\mathbf{H}}$ as the reconstructed hypergraph, we have

$$\hat{\mathbf{H}} = \mathrm{sigmoid}(\mathbf{X}_v^{(L)} \cdot \mathbf{X}_e^{(L)\top}), \tag{13}$$

The hypergraph reconstruction loss is

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{V}| \times |\mathcal{E}|} \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{E}|} (h_{ij} - \hat{h}_{ij})^2 \tag{14}$$

where $h_{ij} \in \mathbf{H}$ and $\hat{h}_{ij} \in \hat{\mathbf{H}}$, respectively, indicate if the i-th node is in the j-th hyperedge in the original hypergraph and the reconstructed hypergraph.

Compared with previous hypergraph learning methods on HKGs, our method has two advantages. First, our hypergraph encoder can extract high-order information among individual H-Fact. Second, we consider node diversity to calculate the attention coefficient, which is ignored in the previous methods.

2.4 Link prediction

Given the input, *i.e.,* an H-Fact with missing elements, we first construct an intra-hypergraph and an inter-hypergraph, and the initialize them to $\mathcal{G}_{inter}(\mathbf{X}_{v,inter}^{(0)}, \mathbf{X}_{e,inter}^{(0)})$ and $\mathcal{G}_{intra}(\mathbf{X}_{v,intra}^{(0)}, \mathbf{X}_{e,intra}^{(0)})$, respectively, After hypergraph encoder, we obtain the final representations of inter-hypergraph $\mathcal{G}_{inter}(\mathbf{X}_{v,inter}^{(L)}, \mathbf{X}_{e,inter}^{(L)})$ and intra-hypergraphs $\mathcal{G}_{intra}(\mathbf{X}_{v,intra}^{(L)}, \mathbf{X}_{e,intra}^{(L)})$. We then extract two node representation vectors corresponding to the missing element in an H-Fact from intra-hypergraph and inter-hypergraph, respectively, *i.e.,* $\mathbf{x}_{intra} \in \mathbf{X}_{v,intra}^{(L)}$ and $\mathbf{x}_{inter} \in \mathbf{X}_{v,inter}^{(L)}$. We further fuse them, followed by employing an MLP to conduct link prediction:

$$p = \mathrm{Softmax}(\mathrm{MLP}(\mathrm{Concat}(\mathbf{x}_{intra}, \mathbf{x}_{inter}))) \tag{15}$$

where p is the probability distribution of the missing element. Moreover, we use the cross-entropy loss between the prediction and the ground truth as

our link prediction loss, *i.e.,*

$$\mathcal{L}_{link} = \frac{1}{S}\sum_{i=1}^{S} y_i \log p_i \tag{16}$$

where p_i is the i-th entry of the prediction p, S is the number of samples, and y_i is the i-th entry of the label y. Finally, the objective function of the proposed method DHE is formulated as:

$$\mathcal{L} = \mathcal{L}_{link} + \mathcal{L}_{rec}^{inter} + \mathcal{L}_{rec}^{intra} \tag{17}$$

where $\mathcal{L}_{rec}^{intra}$ and $\mathcal{L}_{rec}^{inter}$, respectively, are the intra-hypergraph and inter-hypergraph reconstruction loss, from Eq. (14).

3 Experiments

Datasets. The used datasets include three standard H-Fact link prediction benchmarks, *i.e.,* JF17K (Wen et al., 2016), WikiPeople (Guan et al., 2019), and WD50K (Galkin et al., 2020)[1].

Comparison methods. The comparison methods include four transferred methods (*i.e.,* m-TransH (Wen et al., 2016), RAE (Zhang et al., 2018), NaLP (Guan et al., 2019) , and NeuInfer (Guan et al., 2020)) and seven combinated methods (*i.e.,* HINGE (Rosso et al., 2020), Transformer (Galkin et al., 2020), StarE (Galkin et al., 2020), HyperFormer (Hu et al., 2023), ShrinkE (Xiong et al., 2023), GRAN (Wang et al., 2021), and HAHE (Luo et al., 2023)).

Setting-up. We set $L = 6$, $T = 4$, $d_v = 512$, and $d_e = 512$ for our hypergraph Transformer encoder, and set AdamW (Loshchilov and Hutter, 2019) as the optimizer, the learning rate as $3e^{-4}$ and the batch size as 1024. We set the comparison methods according to the original paper to make them output the best results.

Metrics. We follow (Galkin et al. (2020); Wang et al. (2021)) to evaluate all methods by MRR and Hits@k ($k = 1, 10$) under two prediction tasks: (1) predicting an entity (*i.e.,* can be subject or object or value in the qualifier) in an H-Fact, *i.e.,* All Entities for short. (2) predicting either subject or object in an H-Fact, *i.e.,* Subject/Object for short.

3.1 Result analysis

We list the results of all methods on three datasets in Table 1. First, our method achieves the best performance, followed by HAHE, ShrinkE, GRAN, HyperFormer, StarE, Transformer, HINGE, NeuInfer, NaLP, RAE, and m-TransH. For example, our method improves on average by 7.18% and 48.75%, respectively, compared to the best comparison method (*i.e.,* HAHE) and the worst comparison methods (*i.e.,* m-TransH), on all cases. This indicates that it is reasonable to construct both inter-hypergraph and intra-hypergraph by considering one-to-many relation for conducting link prediction tasks on HKGs. Second, the methods (*i.e.,* our method, HAHE, and HyperFormer) consider both the relation among H-Facts and the relation within individual H-Fact for representation learning, but our method outperforms them on all cases. The indicates that our method is the most effective for capturing the relation among H-Facts (by the inter-hypergraph) and the high-order information within individual H-Fact (by the intra-hypergraph).

3.2 Ablation study

Impact of dual-hypergraph. Our DHE designs an inter-hypergraph (*i.e.,* $\mathcal{G}_{intra}$) to capture the ralation among H-Facts and an intra-hypergraph (*i.e.,* $\mathcal{G}_{inter}$) to extract high-order information within individual H-Facts. To demonstrate the effectiveness of each hypergraph, we investigate different combinations of these hypergraphs by reporting the results in Table 2. First, our DHE with inter-hypergraph and intra-hypergraph achieves the best performance. For example, DHE has an average improvement of 15.76% and 1.65%, respectively, compared to DHE w/o inter, and DHE w/o intra, indicating that intra-hypergraph and inter-hypergraph are complement to each other.

Effectiveness of hypergraph decoder. To verify the effectiveness of the hypergraph decoder, we report our method with *i.e.,* DHE w/o decoder, in Table 2. As a result, our method on average increases by 1.71%, compared to DHE w/o decoder. This indicates that the reconstruction loss improves the model predictive accuracy and refines its understanding of H-Facts.

Effectiveness of node diversity. Node diversity is used to measure the contributions of the elements of every H-Fact on HKGs. We report the results of our method and the DHE w/o diversity in Table 2. Obviously, different elements have different contributions. Hence, it is reasonable to set node diversity for link prediction on HKGs.

Effectiveness of H-fact adjacency. Definition 2 is designed to measure the interconnectedness between two H-Facts for constructing the inter-

[1]Related work, the details of datasets, Figures 3-5, and experiments of one-to-many relation can be found in Appedix.

Model	JF17K						Wikipeople						WD50K					
	All Entities			Subject/Object			All Entities			Subject/Object			All Entities			Subject/Object		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
m-TransH	0.102	0.069	0.168	0.206	0.206	0.462	-	-	-	0.063	0.063	0.300	-	-	-	-	-	-
RAE	0.310	0.219	0.504	0.215	0.215	0.466	0.172	0.102	0.320	0.058	0.058	0.306	-	-	-	-	-	-
NaLP	0.366	0.290	0.516	0.221	0.165	0.331	0.338	0.272	0.466	-	-	-	0.224	0.158	0.330	-	-	-
NeuInfer	0.473	0.397	0.618	0.449	0.361	0.624	0.333	0.259	0.477	-	-	-	0.228	0.162	0.341	0.243	0.176	0.377
HINGE	0.517	0.436	0.675	0.431	0.342	0.611	0.350	0.282	0.467	0.342	0.272	0.463	0.232	0.164	0.343	-	-	-
Transformer	-	-	-	0.512	0.434	0.665	-	-	-	0.469	0.403	0.586	-	-	-	0.264	0.194	0.401
StarE	0.542	0.454	0.685	0.574	0.496	0.725	0.378	0.265	0.542	-	-	-	-	-	-	0.349	0.271	0.496
HyperFormer	0.664	0.601	0.787	-	-	-	0.473	0.361	**0.646**	-	-	-	0.366	0.288	0.514	-	-	-
GRAN	0.656	0.582	0.799	0.617	0.539	0.770	0.479	0.410	0.604	0.461	0.370	0.594	0.370	0.302	0.500	0.336	0.266	0.470
ShrinkE	-	-	-	0.589	0.506	0.749	-	-	-	0.485	0.431	0.601	-	-	-	0.345	0.271	0.496
HAHE	0.668	0.597	**0.816**	0.623	0.554	**0.806**	0.495	0.420	0.631	0.509	0.447	**0.639**	0.402	0.327	0.546	0.368	0.291	0.516
DHE	**0.692**	**0.648**	0.781	**0.710**	**0.676**	0.778	**0.562**	**0.537**	0.614	**0.576**	**0.558**	0.609	**0.535**	**0.515**	**0.574**	**0.522**	**0.507**	**0.555**

Table 1: Results of all methods on three datasets. The best scores are highlighted in **bold**, the second-best scores are underlined, and '-' indicates that the results are not reported in previous work.

Model	JF17K						Wikipeople						WD50K					
	All Entities			Subject/Object			All Entities			Subject/Object			All Entities			Subject/Object		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
DHE w/o inter	0.641	0.567	**0.787**	0.601	0.523	0.756	0.437	0.329	0.610	0.436	0.330	0.603	0.374	0.300	0.515	0.342	0.267	0.485
DHE w/o intra	0.681	0.636	0.769	0.696	0.662	0.763	0.538	0.511	0.578	0.549	0.531	0.582	0.522	0.503	0.567	0.510	0.501	0.547
DHE w/o diversity	0.672	0.628	0.760	0.688	0.654	0.757	0.554	0.536	0.614	0.574	0.545	0.608	0.533	0.513	0.573	0.513	0.506	0.553
DHE w/o encoder	0.654	0.610	0.742	0.678	0.644	0.746	0.557	0.525	0.606	0.566	0.543	0.608	0.531	0.506	0.568	0.519	0.500	0.545
DHE	**0.692**	**0.648**	0.781	**0.710**	**0.676**	**0.778**	**0.562**	**0.537**	**0.614**	**0.576**	**0.558**	**0.609**	**0.535**	**0.515**	**0.574**	**0.522**	**0.507**	**0.555**

Table 2: Ablation study of our method on three datasets.

hypergraph, so we investigate its influence in our method by 1) comparing our method only considering the main triple as the adjacency criterion (*i.e.,* "*sro*" for short) with our method considering both the main triple and the qualifiers as the adjacency criterion (*i.e.,* "*sro+av*" for short), and 2) varying the value of k in the range of $\{1, 5, 10, 15, 25, 30\}$. We report the corresponding results in Figures 3 and 4 of Appendix.

Figure 3 shows that "*sro*" is similar to "*sro+av*" in case of All Entities but outperforms "*sro+av*" in case of Subject/Object. This implies that only considering the main triple as the adjacency criterion enhances the predictive performance for Subject/Object, but it does not affect the prediction effectiveness for All Entities. Hence, it is necessary to set the main triple as the adjacency criterion.

Figure 4 clearly demonstrates that the performance of our method first increases to a peak (*i.e.,* k = 10) and then begins to be stable with the increase of the value of k. This indicates that k = 10 is a trade-off between the performance and the computation efficiency of our method.

3.3 Discussion

Besides evaluating the effectiveness of our method in Section 3.1 by demonstrating its effectiveness under the datasets with fixed ratio of H-Facts, *i.e.,* JF17K, WikiPeople, and WD50K, respectively, containing 45.9%, 11.6%, and 13.6% of the H-Facts, in this section, we follow (Xiong et al., 2023; Galkin et al., 2020) to investigate the effectiveness of our method under different ratios of H-Facts. Specifically, we follow (Galkin et al., 2020) to split the dataset WD50K into three datasets, namely, WD50K (33), WD50K (66), and WD50K (100), respectively, containing 33%, 66%, and 100% of the H-Facts. With the same experimental setup in Section 3.1, we list the results of all methods in Figure 5 of Appendix.

Figure 5 shows that our method achieves the best results, followed by HAHE, GRAN, ShrinkE, StarE, Transformer, HINGE, and NaLP. For example, our method improves on average by 10.55%, 3.85%, and 1.45% on WD50K (33), WD50K (66), and WD50K (100), respectively, compared to the best comparison method (*i.e.,* HAHE). This indicates that our method efficiently processes datasets with different ratios of H-Facts.

4 Conclusion

In this paper, we proposed a new combinated HKG method to consider the one-to-many relation among H-Facts and the one-to-many relation within individual H-Fact. Experimental results on three datasets demonstrate the effectiveness of the proposed method for the link prediction tasks of HKGs. The proposed DHE is designed to deal with static HKGs by neglecting the temporal evolution of entities and relations. Future work will focus on handling temporal HKGs.

References

Zirui Chen, Xin Wang, Chenxu Wang, and Zhao Li. 2023. PosKHG: A position-aware knowledge hypergraph model for link prediction. *Data Science and Engineering*, 8(2):135–145.

Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. 2021. Heterogeneous hypergraph variational autoencoder for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4125–4138.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2021. Knowledge hypergraphs: prediction beyond binary relations. In *IJCAI*, pages 2191–2197.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. In *EMNLP*, pages 7346–7359.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2020. NeuInfer: Knowledge inference on N-ary facts. In *ACL*, pages 6141–6151.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021. Link prediction on N-ary relational data based on relatedness evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):672–685.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on N-ary relational data. In *WWW*, pages 583–593.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, pages 1–6.

Zhiwei Hu, Víctor Gutiérrez-Basulto, Zhiliang Xiang, Ru Li, and Jeff Z Pan. 2023. HyperFormer: Enhancing entity and relation interaction for hyper-relational knowledge graph completion. In *CIKM*, pages 803–812.

Rui Li, Xu Chen, Chaozhuo Li, Yanming Shen, Jianan Zhao, Yujing Wang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, et al. 2023. To copy rather than memorize: A vertical learning paradigm for knowledge graph completion. In *ACL*, page 6335–6347.

Yu Liu, Quanming Yao, and Yong Li. 2020. Generalizing tensor decomposition for N-ary relational knowledge bases. In *WWW*, pages 1104–1114.

Yu Liu, Quanming Yao, and Yong Li. 2021. Role-aware modeling for N-ary relational knowledge bases. In *WWW*, pages 2660–2671.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*, pages 1–9.

Yuhuan Lu, Bangchao Deng, Weijian Yu, and Dingqi Yang. 2023. HELIOS: Hyper-relational schema modeling from knowledge graphs. In *ACMMM*, pages 4053–4064.

Haoran Luo, Yuhao Yang, Yikai Guo, Mingzhi Sun, Tianyu Yao, Zichen Tang, Kaiyang Wan, Meina Song, Wei Lin, et al. 2023. HAHE: Hierarchical attention for hyper-relational knowledge graphs in global and local level. In *ACL*, pages 8095–8107.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *WWW*, pages 1885–1896.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, page 6000–6010.

Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021. Link prediction on N-ary relational facts: A graph-based approach. In *ACL*, pages 396–407.

Jiyao Wei, Saiping Guan, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2023. Few-shot link prediction on N-ary facts. *arXiv preprint arXiv:2305.06104*, pages 1–11.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *IJCAI*, pages 1300–1307.

Bo Xiong, Mojtaba Nayyer, Shirui Pan, and Steffen Staab. 2023. Shrinking embeddings for hyper-relational knowledge graphs. In *ACL*, page 13306–13320.

Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *WWW*, pages 1185–1194.

Peng Zhou, Zongqian Wu, Xiangxiang Zeng, Guoqiu Wen, Junbo Ma, and Xiaofeng Zhu. 2023. Totally dynamic hypergraph neural network. In *IJCAI*, page 2476–2483.

A Related work

Based on the ways of the hyper-relational facts definition, previous HKG methods can be grouped into two categories, *i.e.,* transferred HKG methods and combinated HKG methods.

A.1 Transferred HKG methods

Previous works transfer an H-Fact into either an n-tuple (*aka.* n-ary fact) or a set of key-value pairs to learn latent representations of entities and relations of the H-Fact. For example, Wen et al. (2016) proposed m-TransH method, which transfers the H-Fact to an n-tuple, *i.e.,* $r(v_1, v_2, \cdots, v_n)$, to extract embeddings of all entities from a relation-specific hyperplane. On the basis of this work, RAE (Zhang et al., 2018) , HsimplE (Fatemi et al., 2021), and GETD (Liu et al., 2020) further optimize the HKG embedding method. In addition, Guan et al. (2019) directly transfers H-Facts into a set of key-value pairs and proposes the NaLP method to obtain the embeddings of the key-value pairs using a convolution neural network. RAM (Liu et al., 2021) and NaLP+ (Guan et al., 2021) improve NaLP by identifying the importance of roles in H-Facts and considering type information, respectively. However, these transferred HKG methods equally treat each qualifier in H-Fact and thus easily destroy the original structural information of H-Facts.

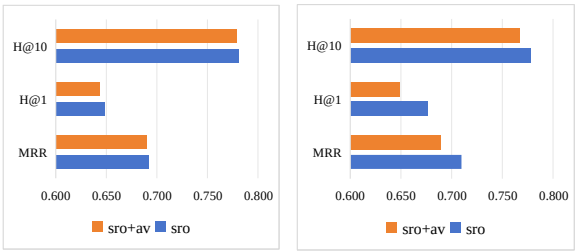
A.2 Combinated HKG methods

Different from transferred HKG methods, the combinated HKG methods combine the main triple with qualifiers to obtain a tuple for directly learning representation from H-Fact. For example, Guan et al. (2020) and Rosso et al. (2020) defined an H-Fact as $\{(s, r, o), \{a_i, v_i\}_{i=1}^{m}\}$. Based on this definition, GRAN (Wang et al., 2021) constructed a heterogeneous graph for an H-Fact, which employs a graph Transformer approach to capturing representations of all elements in the H-Fact. Xiong et al. (2023) proposed a geometric hyper-relational KG embedding method, *i.e.,* ShrinkE, to explicitly model essential inference patterns of H-Facts. Despite this, it ignores the information of global-level structure. Thus, HAHE (Luo et al., 2023) further improves it by integrating the global-level and local-level information to extract representations.

However, existing HKG methods still have limitations to be addressed. First, they ignore the one-to-many relationship between attributes and values in H-Facts, so that they are difficult to capture crucial correlation information. Second, it is difficult for previous methods to capture high-order information in H-Facts based on main triples and qualifiers. To address this, we propose a dual-hypergraph model for representation learning with H-Fact in HKGs.

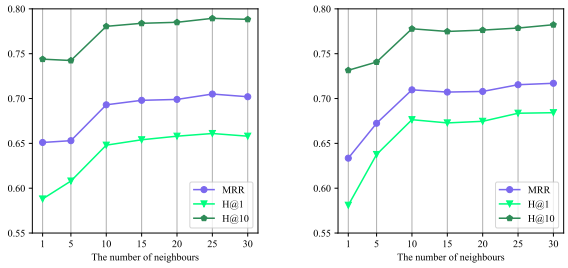
Model	Wikipeople			WD50K		
	MRR	H@1	H@10	MRR	H@1	H@10
GRAN	0.546	0.482	0.643	0.671	0.614	0.752
HAHE	0.464	0.379	0.601	0.640	0.561	0.758
DHE	**0.567**	**0.518**	**0.714**	**0.716**	**0.692**	**0.759**

Table 3: The results of three methods with one-to-many relation on datasets Wikipeople and WD50K.



(a) All Entities (b) Subject/Object

Figure 3: Adjacency analysis of our method on the JF17K dataset.



(a) All Entities (b) Subject/Object

Figure 4: Results of our method with different neighbour numbers on the JF17K dataset.

B Datasets details

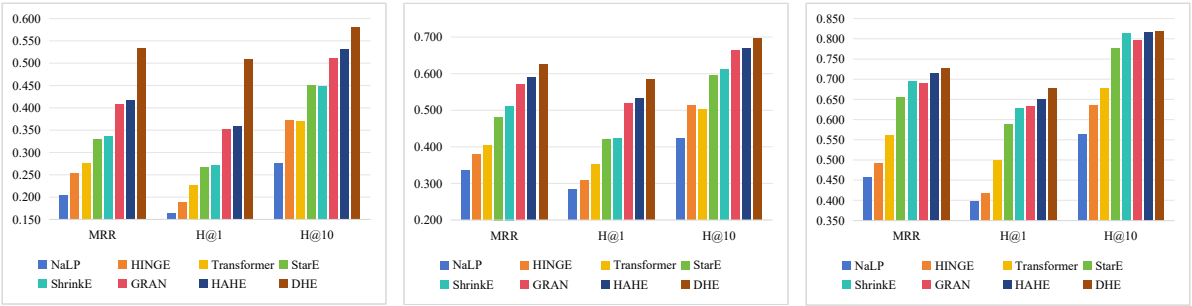
The details of used datasets are listed in Table 4.

C Experiments of One-to-many relation

To verify the effectiveness of the DHE method in handling one-to-many relation, we extract H-Fact data containing one-to-many relation from datasets Wikipeople and WD50K. The experimental results are shown in Table 3, indicating that our proposed DHE method achieves the best performance.

Dataset	All facts	Hyper-relational facts(%)	Entities	Relations	Train	Valid	Test
JF17K	100,947	46320 (45.9%)	28,645	501	76,379	-	24,568
Wikipeople	382,229	44315 (11.6%)	47,765	193	305,725	38,223	38,281
WD50K	236,507	32167 (13.6%)	47,156	532	166,435	23,913	46,159
WD50K(33)	102,107	31866 (31.2%)	38,124	475	73,406	10,568	18,133
WD50K(66)	49,167	31696 (64.5%)	27,347	494	35,968	5,154	8,045
WD50K(100)	31,314	31314 (100%)	18,792	279	22,738	3,279	5,297

Table 4: Data statistics of datasets, where the columns respectively indicate the number of all facts, hyper-relational facts with the number of qualifiers $m > 0$, entities, relations, facts in train/dev/test sets, and all possible arities.



(a) WD50K(33) (b) WD50K(66) (c) WD50K(100)

Figure 5: Link prediction results of different methods on three WD50K datasets.

Compared with the GRAN method, which transfers an H-Fact into a heterogeneous graph, the DHE average increases by 4.29% and 4.31% on the two datasets, respectively, indicating that the regular graph is not enough to characterize such one-to-many relation in H-Facts. In addition, Compared with the HAHE method, which considers global and local level information, DHE has an average improvement of 11.85% and 6.9%, respectively.

Overall, our proposed DHE handles one-to-many relation between H-Fact attributes and values efficiently.

D Limitations

The proposed DHE is designed to deal with static HKGs by neglecting the temporal evolution of entities and relations. Future work will focus on handling temporal HKGs.