

# Understanding the Embedding Models on Hyper-relational Knowledge Graph

Yubo Wang  
 ywangnx@connect.ust.hk  
 HKUST  
 Hong Kong, China

Haoyang Li  
 haoyang-comp.li@polyu.edu.hk  
 The Hong Kong Polytechnic University  
 Hong Kong, China

Shimin Di  
 shimin.di@seu.edu.cn  
 Southeast University  
 Nanjing, China

Fei Teng  
 fteng@connect.ust.hk  
 HKUST  
 Hong Kong, China

Zhili Wang  
 zwangeo@connect.ust.hk  
 HKUST  
 Hong Kong, China

Hao Xin  
 haoxin@tencent.com  
 Tencent  
 Guangzhou, China

Lei Chen  
 leichen@cse.ust.hk  
 HKUST & HKUST(GZ)  
 Guangzhou, China

## ABSTRACT

Recently, Hyper-relational Knowledge Graphs (HKGs) have been proposed as an extension of traditional Knowledge Graphs (KGs) to better represent real-world facts with additional qualifiers. As a result, researchers have attempted to adapt classical Knowledge Graph Embedding (KGE) models for HKGs by designing extra qualifier processing modules. However, it remains unclear whether the superior performance of Hyper-relational KGE (HKGE) models arises from their base KGE model or the specially designed extension module. Hence, in this paper, we data-wise convert HKGs to KG format using three decomposition methods and then evaluate the performance of several classical KGE models on HKGs. Our results show that some KGE models achieve performance comparable to that of HKGE models. Upon further analysis, we find that the decomposition methods alter the original HKG topology and fail to fully preserve HKG information. Moreover, we observe that current HKGE models are either insufficient in capturing the graph's long-range dependency or struggle to integrate main-triple and qualifier information due to the information compression issue. To further justify our findings and offer a potential direction for future HKGE research, we propose the FormerGNN framework. This framework employs a qualifier integrator to preserve the original HKG topology, and a GNN-based graph encoder to capture the graph's long-range dependencies, followed by an improved approach for integrating main-triple and qualifier information to

mitigate compression issues. Our experimental results demonstrate that FormerGNN outperforms existing HKGE models.

## CCS CONCEPTS

- Computing methodologies → Knowledge representation and reasoning;

## KEYWORDS

Hyper-relational Knowledge Graph, Link Prediction, Graph Neural Network

### ACM Reference Format:

Yubo Wang, Shimin Di, Zhili Wang, Haoyang Li, Fei Teng, Hao Xin, and Lei Chen. 2018. Understanding the Embedding Models on Hyper-relational Knowledge Graph. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 INTRODUCTION

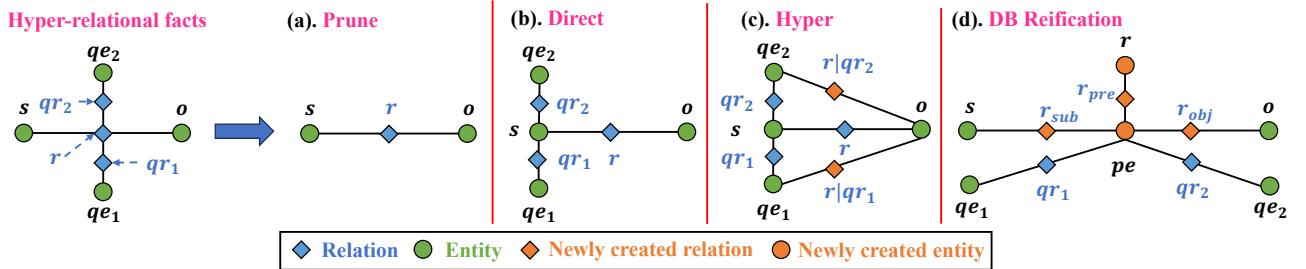
Knowledge Graphs (KGs) [20, 35] are powerful tools for organizing and leveraging human knowledge. They have facilitated a wide range of web applications, including question answering [16, 41, 49], semantic search [17, 27, 40], and recommendation systems [11, 23, 45, 47]. Similar to the prune decomposition method shown in Figure 1-(a), KGs store factual information in the form of triplets  $(s, r, o)$ , where entities  $s$  and  $o$  are connected by a relation  $r$ . However, this triplet format has been criticized for being insufficient and prone to information loss [8, 21, 22, 24, 26, 28, 38]. To address this limitation, researchers have extended KGs by incorporating hyper-relational facts. These are represented by adding a qualifier set  $Q$ , which contains extra entity-relation pairs  $(qr, qe)$  to more accurately reflect real-world facts, leading to the development of Hyper-relational KGs (HKGs). An example of the hyper-relational fact is shown in Figure 2, where the qualifiers with relations Degree and Major helps to identify the each university that Albert Einstein

---

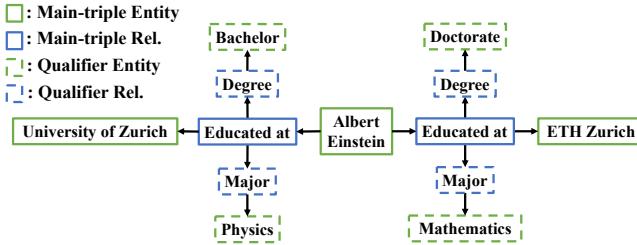
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-XXXX-X/2018/06...\$15.00  
<https://doi.org/XXXXXXX.XXXXXXX>



**Figure 1: Illustration of our 3 decomposition methods (sub-figure (a) to (c)) and the DB reification method by [2] (sub-figure (d)). In (c),  $r|qr_1$  and  $r|qr_2$  represents new relationships constructed from the main relation  $r$  and qualifier relation  $qr$ . In (d),  $r_{pre}$ ,  $r_{sub}$ ,  $r_{obj}$  represents the newly generated relationships, and  $pe$  represents the newly generated pseudo entity.**



**Figure 2: An HKG example, without the qualifiers with relation Degree and Major, the difference between entity University of Zurich and ETH Zurich w.r.t. Albert Einstein is hard to detect, as Einstein were educated at both of these universities.**

graduated from. Given HKGs' practicalities, they were widely applied in many real-world applications. For example, HKGs enable reasoning over time-dependent facts on temporal graphs by incorporating temporal qualifiers into relationships [6], aiding applications like historical event analysis [14] or forecasting [13].

Given the importance of HKGs, researchers have focused on developing embedding models specifically for them, known as Hyper-relational Knowledge Graph Embedding (HKGE) models. Most current HKGE models are extensions of existing KG embedding (KGE) models. For instance, the well-known GNN-based HKGE model StarE [8] is built upon KGE model CompGCN [33].

Since most classic HKGE models are extended from KGE models, this inevitably raises a question: *Are their effectivenesses because of their KGE base model, or the specially designed qualifier processing module?* Although pioneer work [2] has found specially designed HKGE models are sometimes not better than KGE with the simple DB reification method. It claims that this phenomenon shows the robustness of their specially designed decoder modules, while ignoring KGE base models. Furthermore, it only focuses on one GNN-based KGE model CompGCN [33] and its HKG extension StarE [8]. With the development of the KGE and HKGE research, various novel models with new technical routines are proposed [15, 50], do these novel HKGE models also perform similarly with more recent KGE models? Therefore, we dig into the performance of KGE and HKGE models on HKGs to answer this question.

To make fair comparisons between KGE and HKGE models on HKGs, we need to make KGE models runnable on HKGs. Previous researchers apply model-wise extension on KGE model, which

focus on upgrading KGE models to model HKG [7, 8, 25, 28, 30, 36, 38]. In this paper, we turn to another view to data-wise extend HKG to KG format. As shown in Figure 1-(d), DB reification [2] converts main-triple relation into entity and then add pseudo nodes and relations to represent hyper-relational facts. It cannot preserve the most important main triplets, and hinders us from incrementally studying the effect of each HKG component. Consequently, we propose three decomposition methods that can convert HKG to KG's triplet format while keeping the original format of HKG main-triples and preserving the HKG information to different extents. In this paper, we divide the preserved HKG information into 3 parts: main triple, intra-qualifier, and inter-qualifier information (The detailed definition of them are in subsection 3.1). Through such a data-wise way, we can easily test conventional KGE models on HKGs. As discussed in section 3, our preliminary experimental results demonstrate that KGE models can still achieve comparable performance to those specially designed HKGE models on HKGs.

As the innovation of most of the HKGE model is their specially designed qualifier capturing modules, two questions naturally arise from the results:

- (1) *Is it necessary to develop various HKGEs?*
- (2) *What information are current HKGEs overlooking?*

In section 4, we further delve deeper into the model performance comparison between KGE and HKGE models to answer these two questions. Firstly, we examine how these models capture qualifiers, as the key distinction between KGE and HKGE models lies in the inclusion of specialized modules designed to handle qualifiers. Our findings show that HKGE models are still necessary, as KGE models with decomposition methods shift the original HKG topology and are inadequate for capturing qualifiers in HKGs. As a result, on HKG, KGE models suffer from *information loss*, such as semantic shifts and misinterpretations. Secondly, to identify the information missed by HKGE models, we analyze the graph reception approach of both HKGE and KGE models. By “graph reception,” we refer to how models receive and integrate information from the HKG, including both main-triple and qualifier information. Our analysis reveals that current HKGE models often suffer from a *limited receptive field*, which restricts their ability to capture the graph’s multi-hop topology. This limitation weakens their capacity to model the graph’s long-range dependency, which is crucial for many tasks. Furthermore, while some HKGE models have broader

**Table 1: Overview of existing HKG embedding models and how they receive graph topology information and integrate qualifiers.**  $l$  denotes the number of layers;  $\langle \cdot \rangle$  denotes the multi-linear product;  $\mathbb{P}_r$  denotes the projection matrix of relation  $r$ ;  $HT$  denotes the qualifier integration function of the HyTransformer [42]; Other notations can be referred from Table 2.

Category	KGE Model	HKGE Model	Key component modeling HKG	
			Receptive Field	Qualifier Integration Function
Tensor	SimplE [18]	HypE [7]	Fact level ( $\leq 1$ -hop): $F(s, r, Q)$	$\langle h_r, Conv_1(h_s), Conv_2(h_{q_1}), \dots, Conv_{a_r}(h_{q_{a_r-1}}) \rangle$
Decomposition	ComplEx [32]	RAM [25]		$\sum_{i=1}^{a_r} \langle h_r^i, P_r^i[1, :]h_s, P_r^i[2, :]h_{qe_1}, \dots, P_r^i[a_r, :]h_{qe_{a_r-1}} \rangle$
Simple NN	ConvE [5]	NaLP_fix [36]		$MLP(Conv(h_s, h_r, h_{qr_1}, h_{qe_1}, \dots))$
		Hinge [28]		$Cat(Conv_1(h_s, h_r), Conv_2(h_s, h_r, h_{qr_1}, h_{qe_1}, \dots))$
Geometric	TransH [37]	m-TransH [36]		$\sum_{i=1}^{a_r} W_i \mathbb{P}_r(h_i) + b_r$
	-	ShrinkE [39]		$\min_{(qr, qe) \in Q} (\phi(MLP_1(h_r, h_{qr}, h_{qe}), MLP_2(h_r, h_{qr}, h_{qe})))$
Transformer	-	HyTransformer [42]		$Trm(Cat(h_s, h_r, h_{qr_1}, h_{qe_1}, \dots))$
		HyperFormer [15]	Node level ( $= 1$ -hop): $\sum_{r', Q' \in N(s)} (s, r', Q')$	$Conv(HT(s, r, Q), \gamma \frac{\sum_{(s, r', Q') \in N(s)} Trm(F(s, r', Q'))}{ N(s) }, \gamma(h_{qr_1}, h_{qe_1}, \dots))$
		HAHE [26]		$MLP(Cat(Trm(Cat(h_s, h_r, h_{qr_1}, h_{qe_1}, \dots)), h_r, b_e, b_r))$
GNN	CompGCN [33]	StarE [8]	Graph level ( $> 1$ -hop): $(\sum_{r', Q' \in N(s)} (s, r', Q'))^l$	$\phi(h_s, \gamma(h_r, h_{qr_1}, h_{qe_1}, \dots))$
		QUAD [30]		$\gamma(\phi(h_s, h_r), h_{qr_1}, h_{qe_1}, \dots)$

receptive fields, they still compress qualifier information into fixed-sized main triple embedding matrices, leading to the *information compression issue*.

The above experiments are all based on an experimental observation of current works. Therefore, we propose a new framework FormerGNN to incorporate several key mechanisms to justify the above speculations on qualifier capture and graph reception. Firstly, FormerGNN employs a transformer-based qualifier integrator to capture the original topology of the HKG within the 1-hop neighborhood, eliminating the need for decomposition methods. Secondly, it uses a GNN-based graph encoder to maintain a broad receptive field for main triples and effectively capture the graph's long-range dependencies. Finally, FormerGNN performs predictions jointly with the main-triple and qualifier embedding matrix, without the need to compress one into another, mitigating the information compression issue. Experimental results demonstrate that FormerGNN achieves state-of-the-art performance across all HKGE models on the Cleaned JF17k and WD50k benchmarks.

Our contributions can be summarized as follows:

- Although various HKG embedding models have been developed, the source of their notable performance is yet cleared. In this paper, we conduct a comprehensive analysis, to systematically evaluate whether their performance is due to their specially designed qualifier processing module, or their KGE base model.
- To data-wise extend HKG to KG format and analyse the performance of KGE model on HKG, we proposed three HKG decomposition methods that can preserve HKG information to different extents, while maintaining HKG's original main-triple topology.
- To justify our claims while also providing a possible better HKGE research direction, we propose FormerGNN<sup>1</sup>, which leverages a transformer based qualifier integrator that can capture HKG's original topology, a GNN graph encoder to capture the graph's

long range dependency, and predict jointly with all obtained embeddings and mitigate GNNs' information compression issue.

## 2 PRELIMINARY

### 2.1 Further HKG Backgrounds

As illustrated, KG's triplet format alone may suffer from information loss [38], and are hard to represent the facts with more than two entities [28]. Hence, to address the shortcomings of conventional KGs, the facts within these KGs have been extended to hyper-relational facts [8] (Figure 1) by adding multiple key-value pairs named qualifiers, and form HKGs. In this paper, following [8], we denote the hyper-relational facts  $F(s, r, o, Q)$  as a main triplet along the qualifier set which contains  $n$  qualifiers:  $Q := \{(qr_i, qe_i)\}_{i=1}^n$ . For triplet facts, they can be considered hyper-relational facts with the number of qualifiers  $n = 0$ , in this paper, we represent them as subject-relation-object triples:  $(s, r, o)$ , and formally formulate HKGs as:  $\mathcal{G}(\mathcal{V}, \mathcal{R}, \mathcal{F})$ , where  $s, o, qe_1, \dots, qen \in \mathcal{V}$ ,  $r, qr_1, \dots, qr_n \in \mathcal{R}$ , and  $F(s, r, Q) := (s, r, o, \{(qr_i, qe_i)\}_{i=1}^n) \in \mathcal{F}$ , with  $\mathcal{V}$  representing the entity set (node set),  $\mathcal{R}$  representing the relation set (edge set), and  $\mathcal{F}$  representing the fact set. In this paper, we denote scalars by lowercase letters  $r$ , and its respective vector by  $h_r$ .

### 2.2 Specially Designed HKG Embedding Models

In this paper, we categorize current HKGE methods into five categories based on their approach to integrating qualifiers. We will primarily focus on the following HKGE models, which represent the key methodologies in these categories.

**Tensor Decomposition Models** These models decompose the basis matrix into several matrices, to represent entities and relations in KGs and HKGs. For example, RAM [25] is a tensor decomposition model based on the classic bilinear KGE model ComplEx [32]. Compared with ComplEx, it further considers the main triple and qualifiers as different roles of the hyper-relational facts and models them with role-specific pattern matrices. On the other hand, HypE [7]

<sup>1</sup>Github: <https://github.com/Wyb0627/GraphEmbedding>

**Table 2: Summary of Important Notations.**

Notations	Meanings
$s$	The subject entity of main-triple.
$r$	The relation of main-triple.
$qe$	The qualifier entity.
$qr$	The qualifier relation.
$h_*$	The embedding matrix of an entity or a relation.
$W$	The learnable weight matrix.
$b$	The learnable bias.
$\mathcal{G}(\mathcal{V}, \mathcal{R}, \mathcal{F})$	The knowledge graph with node set $\mathcal{V}$ , edge set $\mathcal{R}$ , and fact set $\mathcal{F}$ .
$F(s, r, Q)$	The hyper-relational fact.
$Q$	The qualifier set.
$\mathcal{N}(\cdot)$	The 1-hop graph neighborhood of an entity.
$Conv$	The convolutional operation.
$Cat$	The matrix concatenation operation.
$\phi$	The message passing function.
$\oplus$	The aggregation function.
$\gamma$	The composition operation.
$Trm$	The transformer layers.
$[MSK]$	The MASK token of the transformer.

extends another bilinear KGE model SimpE [18] by considering additional qualifier positions and applying a abstract relation to loosely represent a combination of all qualifier relations of the original hyper-relational fact.

This category of models treats each fact independently and aggregates the information at the end. Consequently, according to Table 1, they only directly receive information within facts and have a receptive field less or equal to the graph’s 1-hop neighborhood.

**Simple NN-based Models** These models apply simple, lightweight neural network structures, such as Convolutional Neural Networks (CNNs) or Multi-layer Perceptrons (MLPs), directly to the feature matrices of facts to learn their semantic information. For example, NaLP [36] and Hinge [28] extend their base KGE model, ConvE [5], by further appending the feature matrices of qualifiers to those of main-triples to create the feature matrices of hyper-relational facts. Then they apply CNN or MLP layers to these feature matrices to encode hyper-relational facts and then aggregate these facts w.r.t. their subject entities and relations for the graph’s entity and relation embeddings.

Similar to Tensor Decomposition Models, Simple NN-based models also treat each fact independently and have receptive field less or equal to the graph’s 1-hop neighborhood.

**Geometric Models** These models embed KGs or HKGs by projecting entities and relations onto hyperplanes and trying to optimize the model based on the geometric cost functions. KGE model TransH [37] only applies the cost function for the main triple. In order to model HKGs, m-TransH [38] extends TransH by considering the separate cost functions for entity-relation pairs within hyper-relational facts and calculating a weighted sum of their cost together. ShrinkE [39] is a model specifically designed to model HKG, it maps main triples to boxes in the geometric hyperplane, then considers the qualifiers as geometric shrinkings of the box and uses this extra information further to reduce the search space for the answer.

As Geometric Models model each fact independently, according to Table 1, they also have a receptive field less or equal to the graph’s 1-hop neighborhood.

**Transformer-based Models** This category of models focuses on applying the dense parameter transformer [34] on feature hyper-relational facts’ feature matrices to integrate qualifiers and the main-triple and directly carry out prediction with the transformer’s last layer output. Among this category, HyTransformer [42] is a simple method that treats hyper-relational fact as a sequence of entities and relations, and does not apply any special approach to deal with qualifiers:

$$\begin{aligned} h_o &= h_{[MSK]} = Trm(F(s, r, [MSK], Q)) \\ &= Trm(Cat(h_s, h_r, h_{[MSK]}, h_{qr_1}, h_{qe_1}, \dots)), \end{aligned} \quad (1)$$

where the embedding generated from the mask token [MSK] is considered as the embedding  $h_o$  of the predict target entity  $o$ , and  $Cat$  denotes the concatenation operation. HyTransformer substitutes the GNN in StarE with Layer Normalization and Dropout layers and only uses the Transformer decoder for prediction, to achieve the fact-level receptive field. Furthermore, a qualifier prediction subtask is applied to enhance its link prediction performance. GRAN [36] treats the hyper-relational fact as a heterogeneous graph and maintains the connectivity between qualifiers through the attention mechanism of the transformer.

Building upon HyTransformer, HyperFormer [15] further incorporates additional aggregators to capture local-level sequential information. It also applies convolutional operations to integrate information from multiple aggregators:

$$\begin{aligned} h_o &= Conv(h_{[MSK]}, \gamma_{rq}(h_{qr_1}, h_{qe_1}, \dots), \\ &\quad \gamma_{en}(\sum_{(s, r', Q') \in \mathcal{N}(s)} Trm(s, r', Q') / |\mathcal{N}(s)|)), \end{aligned} \quad (2)$$

where  $Conv$  denote the convolutional operator,  $h_{[MSK]}$  is the same as in Equation 2.2,  $\gamma_{rq}$  and  $\gamma_{en}$  are composition operations of HyperFormer’s relation qualifier and entity neighbor aggregator, respectively. Consequently, HyperFormer can directly receive graph information within the graph’s 1-hop neighborhood, rather than the fact only.

On the other hand, to increase the model’s graph receptive field, HAHE [26] utilizes the graph transformer [29, 43, 44] to aggregate the HKG’s global dependency into node embeddings, and then model the HKG’s local dependency as bias injected to the hyper-relational fact embedding:

$$MLP(Cat(Cat(h_s, h_r, h_{qr_1}, h_{qe_1}, \dots), h_r, b_e, b_r)). \quad (3)$$

where  $b_e$  and  $b_r$  denotes the entity and relation bias, respectively.

**Graph Neural Network Models** Current Graph Neural Network (GNN)-based KGE models can be classified into two groups: neighbor-aggregated models, and path-aggregated models. Given a KG  $\mathcal{G}(\mathcal{V}, \mathcal{R}, \mathcal{F})$ , neighbor-aggregated KGE model (Neighbor-GNN in Table 3) CompGCN [33] extends GNNs to multi-relational graphs and propose the node representation

$$h_o = act\left(\bigoplus_{(s, r) \in \mathcal{N}(s)} (\{W_{\lambda(r)} \phi(h_s, h_r)\})\right), \quad (4)$$

where  $\bigoplus$  denotes the aggregation function;  $h_o$  and  $h_r$  denote the embedding vector of entity  $o$  and relation  $r$  respectively,  $\phi(\cdot) : \mathbb{R}^d \times$

**Table 3: Performance of KG/HKG embedding methods on two most representative HKG benchmarks.**

Category	KGE Model	HKGE Model / Modules	Receptive Field	Cleaned JF17k [8]			WD50K [8]		
				MRR	H@1	H@10	MRR	H@1	H@10
Tensor Decomposition	SimpleE [18]	HypE [7]	≤ 1-hop	0.295	0.211	0.454	-	-	-
	ComplEx [32]	RAM [25]		0.342	0.257	0.505	0.300	0.236	0.422
	Simple NN	NaLP_fix [36]		0.067	0.044	0.108	0.177	0.131	0.264
		Hinge [28]		0.272	0.178	0.461	0.243	0.176	0.377
	Geometric	TransH [37]		0.262	0.188	0.393	-	-	-
	-	m-TransH [38]		0.343	0.257	0.510	0.323	0.260	0.441
Transformer	TransH [37]	ShrinkE [39]	>1-hop	0.368	0.265	0.592	0.348	0.272	0.489
	-	HyTransformer [42]		0.405	0.310	0.585	0.360	0.288	0.496
		HAHE [26]		<b>0.425</b>	<b>0.328</b>	<b>0.613</b>	<b>0.372</b>	<b>0.293</b>	<b>0.522</b>
Neighbor-GNN	CompGCN [33]	StarE [8]	>1-hop	0.381	0.284	0.577	0.346	0.268	0.494
		QUAD [30]		0.402	0.305	0.602	0.349	0.275	0.487
Tensor Decomposition	ComplEx [32]	Pruned	≤ 1-hop	0.112	0.079	0.173	0.103	0.058	0.194
		Direct		0.103	0.074	0.156	0.102	0.057	0.188
		Hyper		0.111	0.076	0.174	0.134	0.087	0.225
Simple NN	ConvE [5]	Pruned		0.120	0.074	0.210	0.042	0.014	0.096
		Direct		0.131	0.084	0.219	0.033	0.015	0.066
		Hyper		0.138	0.078	0.248	0.041	0.017	0.103
Geometric	TransH [37]	Pruned		0.118	0.031	0.307	0.167	0.008	0.347
		Direct		0.130	0.037	0.335	0.169	0.008	0.352
		Hyper		0.136	0.042	0.345	0.171	0.008	0.354
Path-GNN	NBFNet [50]	Pruned	>1-hop	<b>0.415</b>	<b>0.313</b>	<b>0.630</b>	<b>0.360</b>	0.271	<b>0.530</b>
		Direct		0.412	0.308	0.623	0.351	0.261	0.523
		Hyper		0.396	0.299	0.580	0.346	0.257	0.517
Neighbor-GNN	CompGCN [33]	Pruned		0.383	0.291	0.577	0.348	0.273	0.487
		Direct		0.382	0.290	0.579	0.350	<b>0.277</b>	0.488
		Hyper		0.389	0.294	0.590	0.350	<b>0.277</b>	0.488

$\mathbb{R}^d \rightarrow \mathbb{R}^d$  is the message passing function,  $W_{\lambda(r)}$  is the direction-specific weighting matrix, and  $\mathcal{N}(s)$  denotes the neighborhood of node  $s$ .

Based on CompGCN, HKGE model StarE [8] was proposed. It adapts to HKGs by further generate embeddings  $h_q$  for each qualifier pair  $q$  as the embedding aggregation of the qualifier entity  $h_{qe}$  and qualifier relation  $h_{qr}$ :

$$h_q = W(\sum \{\phi_q(h_{qr}, h_{qe})\}_{(qr, qe) \in Q}). \quad (5)$$

Next, StarE would aggregate  $h_q$  into the main triple's relation embedding matrix  $h_r$ :

$$h_o = act(\bigoplus_{(s,r) \in \mathcal{N}(s)} (\{W_{\lambda(r)}\phi(h_s, \gamma(h_r, h_q))\})), \quad (6)$$

where  $\phi_q$  can be any entity-relation function akin to  $\phi$ ;  $\gamma$  can be any function that combines multiple embeddings, such as the weighted sum or simple average.

Based on StarE, QUAD [30] aggregates the qualifier information into the entity and relation embedding of the main triple:

$$h_o = act(\bigoplus_{(s,r) \in \mathcal{N}(s)} (\{W_{\lambda(r)}\phi(\gamma(h_s, h_r), h_q)\})), \quad (7)$$

where  $\gamma(\cdot)$  denote the weighted sum. For the extra qualifier aggregator, QUAD introduces main triple embedding:

$$h_t = W(Cat(h_s, h_r, h_o)) + b, \quad (8)$$

and treat  $h_t$  the same as entity embeddings to take part in the later message passing, enabling qualifiers to receive information from the main triple.

For the path-aggregated GNN-based KG embedding model (Path-GNN in Table 3) NBFNet [50], a distinct approach is employed. Rather than aggregating the neighborhood of a specific entity on the KG, NBFNet aggregates paths from one entity to another within its multi-hop graph neighborhood. The pair embedding is then created by aggregating path representations between entities  $s$  and  $o$  w.r.t. query relation  $r$ :

$$h_r(s, o) = \bigoplus_{P \in \mathcal{P}(s, o)} (\phi(\prod_{i=1}^{|P|} h_{r_i})), \quad (9)$$

where  $\mathcal{P}(s, o)$  denotes the set of path from entity  $s$  to entity  $o$ ;  $h_{r_i}$  denotes the  $i$ -th relation within path  $P$ . The path-GNN is the most popular technique in current KGE research and has led to the development of many novel models [4, 48].

Benefiting from the message passing mechanism [19], GNN-based models can directly receive graph information within the graph's multi-hop neighborhood.

In this paper, we focus on Cleaned JF17k [8] and WD50k [8], two widely used datasets that have moderate qualifier density and graph scale, and are considered HKGs to have the most representative characteristics. Cleaned JF17k, which was proposed based on the original JF17k [38], further addresses the data leakage issue.

On the other hand, we also conducted experiments with the Wikipeople [10] and FBAUTO [7] datasets, which exhibit distinct HKG characteristics. As shown in Table 4, Wikipeople has a low qualifier density (less than 3%), while FBAUTO is small in graph scale, containing only 2,094 unique entities and 8 unique relations.

### 3 MODEL PERFORMANCE ON HKG

In this section, we experimented the performance of KGE and HKGE models on HKG, to see whether KGE can still perform similarly with HKGE nowadays. Different from most of previous researches, we data-wise decompose HKG into KG format, to analyse the capability of KGE models on HKG. To achieve this, we propose the 3 decomposition methods in subsection 3.1. With the help of these methods, we carry out experiment with KGE models from 5 technical categories, interestingly, in subsection 3.3, we find some KGE can perform comparably with HKGE.

#### 3.1 Decomposition Methods

Although [2] applied the DB reification method to convert HKG to KG format, as introduced, it can greatly shift the original HKG topology, even for the important main triples. As a result, we cannot analyse the HKG's each component with its DB reification method. Hence, to data-wise adapt KGE models to HKG and to better study the impact of different HKG information on KGE models, we design 3 simple decomposition methods that can preserve the HKG information to different extent:

- **Prune method** prunes all qualifiers, only preserves the *main-triple information*:

$$T_{\text{prune}} := \{(s, r, o)\}; \quad (10)$$

- **Direct method** directly links the qualifiers to the subject entity of the main triple:

$$T_{\text{direct}} := \{(s, r, o), \{(s, qr_i, qe_i)\}_{i=1}^n\}, \quad (11)$$

compared with the prune method, it further considers qualifiers the same as main triples and further preserve the *intra-qualifier information*, including qualifier entities and relations.

- **Hyper method** creates new relation  $r||qr_i$  for every qualifier  $(qr_i, qe_i)$  with the name concatenation of relation  $r$  and qualifier relation  $qr_i$ :

$$T_{\text{hyper}} := \{(s, r, o), \{(s, qr_i, qe_i)\}_{i=1}^n, \{(qe_i, r||qr_i, o)\}_{i=1}^n\}. \quad (12)$$

Compared to the direct method, this approach better preserves *inter-qualifier information*, including the relation between qualifiers and the main triple, as well as between qualifiers within the same hyper-relational fact.

Overall, the decomposition methods keep the KGE models' original receptive field and make KGE models aggregate qualifiers similarly with aggregating main-triple information.

#### 3.2 HKG datasets

#### 3.3 Interesting Findings

In this paper, we evaluate several HKGE models alongside KGE models using our simple decomposition methods, as shown in Table 3. The results for NaLP\_fix and Hinge on WD50k are taken from [39],

**Table 4: HKG Dataset Statistics.** “#Qual.” denotes the range of the number of qualifier entity-relation pairs, “Tri.” denotes the triplet facts with no qualifiers, and “HR” denotes the hyper-relational facts.

Dataset	V	R	#Qual.	#Tra.	#Val.	#Tst.	#Tri.	#HR
Cleaned JF17k [8]	25,092	320	0-4	49,120	12,280	17,635	54,551	24,484
WD50k [8]	47,155	531	0-65	166,435	23,913	46,159	204,340	32,167
Wikipeople [10]	31,038	171	0-7	262,301	33,838	33,806	257,693	4,608
FBAUTO [7]	2,094	8	0-3	6,778	2,255	2,180	3,786	7,427

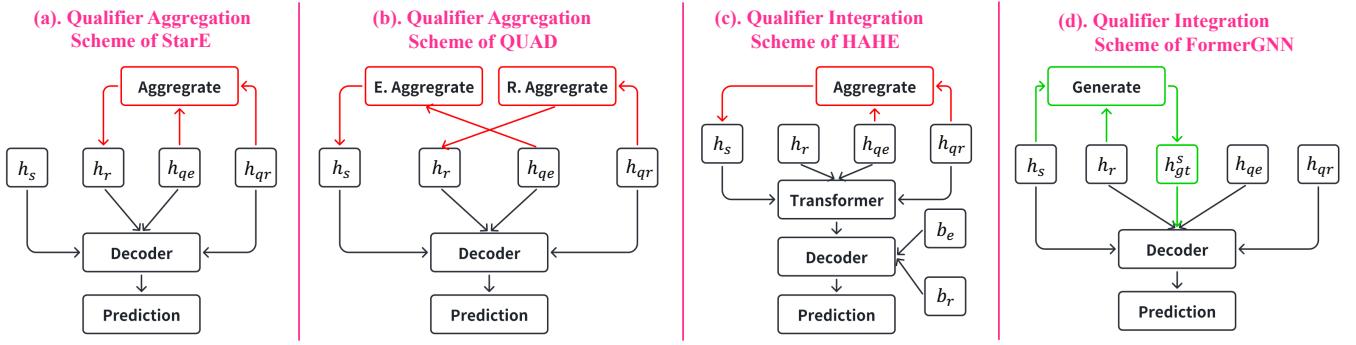
while the results for ComplEx, ConvE, and TransH were reproduced using PyKeen [1]. Other results were obtained from the original GitHub repositories. Since StarE, QUAD, and HyTransformer use the Transformer decoder, we replace the decoder in CompGCN with a Transformer to ensure consistency with these models. All experiments were conducted on a single NVIDIA A100-40GB GPU, and parameters were either kept consistent with the original repositories or tuned using grid search when dataset-specific parameters were not available.

As shown in Table 3, compared to more advanced GNN-based KGE models, Tensor Decomposition (TD)-based KGE models are significantly outperformed by their respective HKGE extensions. On one hand, when applied to the pruned method, the HKG main triples differ from the KG main triples in that they do not contain complete information about HKG facts, as they lack supporting qualifier information. As a result, when coping TD-based KGE models with HKG decomposition methods, the main-triple embeddings were affected by the information that should exist in the qualifiers. Hence, the decomposition from the basis matrix into the main triple's embedding makes no sense and can severely bring noise. On the other hand, the GNN-based KGE models aggregate the respective graph neighborhood for generating main triple embeddings. In that case, the qualifier information just missing, rather than being simply injected into the main triple embedding and serving as noise. Hence, the negative effect of the prune decomposition method is less severe. Furthermore, since GNN-based KGE models separate their encoder and decoder modules, their performance can benefit from HKG-oriented decoders [2] when using the direct and hyper methods. This makes GNN-based KGE models the most effective choice for applying to HKGs.

More importantly, when applying the three decomposition methods, the KGE models CompGCN [33] and NBFNet [50] achieve performance that is comparable to, or even better than, some popular HKGE models, such as CompGCN's HKG extension StarE [8]. These results introduce a need to further analyze the strengths and weaknesses of some of the best-performed KGE and HKGE models, and raise two questions.

- (1) Consider the key difference between KGE and HKGE is whether they apply specially designed modules to capture qualifiers, is it necessary to develop various HKGEs?
- (2) Why would current HKGE models not perform better? What important information are they overlooking?

To answer these questions, we study current KG and HKG models regarding their qualifier capture and graph reception ability in the next section.



**Figure 3:** The qualifier aggregation scheme of StarE, QUAD, and HAHE compared with our FormerGNN framework, where E. Aggregate and R. Aggregate represent QUAD’s qualifier entity and relation aggregation operation, respectively. Compared with StarE, QUAD, and HAHE, FormerGNN does not aggregate qualifier information in  $h_{qe}$  and  $h_{qr}$  into the main-triple’s embeddings  $h_s$  or  $h_r$ . Instead, FormerGNN stores the graph’s long-range dependency information separately in  $h_{gt}^s$ , then jointly and directly carries out prediction with  $h_{gt}^s$ , and information from the main-triple and qualifiers embedding matrixes.

**Table 5: HKG information preserved by different decomposition methods.**

Method	Preserved Information		
	Main Triple	Intra-Qualifier	Inter-Qualifier
Prune	✓		
Direct	✓	✓	
Hyper	✓	✓	✓
DB reification [2]		✓	✓

## 4 QUALIFIER CAPTURE AND GRAPH RECEPTION

As introduced, to answer two questions raised in the previous section, we need to analyze current KGE and HKGE models’ ability from two parts: Qualifier Capture and Graph Reception.

- In subsection 4.1, after analyzing the performance of several best-performed KGE models with the three decomposition methods, we claim it is still necessary to develop more novel HKGEs, as the ability of decomposition methods to preserve the HKG information is still insufficient.
- In subsection 4.2, after analyzing the graph reception ability of current HKGE models, we find these models are either insufficient in capturing the graph’s long-range dependency or suffer from information compression issues when aggregating qualifiers with the main triple.

### 4.1 Qualifier Capture

In this section, we analyze the ability of KGE and HKGE models to capture qualifiers to address the question of whether it is necessary to develop various HKGEs. Based on the following observations, we argue that HKGEs are still needed, as the decomposition methods are insufficient in integrating qualifiers.

- Firstly, the path-GNN KGE model NBFNet, when preserving more HKG information, can suffer from a performance drop on Cleaned JF17k and WD50k.
- Secondly, for neighbor-GNN KGE model CompGCN, when preserving more information, its performance remains similar or only marginally increase on two most representative dataset.

- Thirdly, the Transformer-based HKGE model HyperFormer achieves the SOTA performance on two most representative dataset, this also shows the importance of a properly and specially designed qualifier capturing mechanism, to learn the qualifier together with the main triple, rather than inject one into another.

These phenomena occur because qualifiers are often contain less important information than the main triple [28]. Hence, the direct and hyper methods may introduce noise paths or neighborhoods formed by qualifiers, which can interfere with capturing the more important main-triple information.

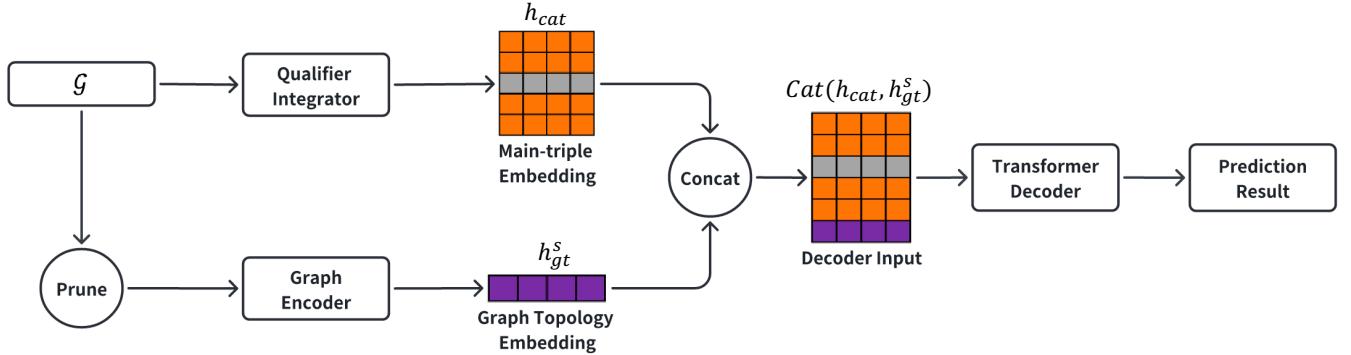
To further support our claim, we enhance the qualifier capture approach of the top-performing KGE model, NBFNet, resulting in an improved model called FormerGNN. FormerGNN integrates a Transformer-based qualifier capturing module with NBFNet to capture the original HKG topology. A more detailed analysis of FormerGNN are in section 5.

### 4.2 Graph Reception

To point out what important information are current HKGE models overlooking, in this section, we analyze current KGE and HKGE models’ graph reception ability. With the following observations from Table 3, we claim some of the current HKGEs overlook the capture graph’s long-range dependency.

- Firstly, with an additional GNN graph encoder capturing the graph’s long-range dependency, QUAD [30], and even CompGCN [33] achieves better performance than HyTransformer [42] on Cleaned JF17k and WD50k, which does not align with the claim of previous researches [15, 42].
- Secondly, by employing the latest path-GNN KGE technique, NBFNet [50] is the best at capturing the graph’s long-range dependencies among all tested KGE models. As a result, it achieves the second-best performance on Cleaned JF17k and WD50k, even when using the prune method and lack of qualifier information.

We claim the reason behind these results as these models only achieve a limited receptive field. Specifically, a wide receptive field is beneficial for capturing the graph’s long-range dependencies, which in turn contribute to improved model performance. Although



**Figure 4:** The pipeline the FormerGNN framework, it involves concatenating the main-triple and qualifier embeddings, generated by the Qualifier Integrator with the original HKG topology, and the graph topology embeddings generated by the Graph Encoder with the HKG after applying the prune decomposition method.

**Table 6:** The performance of FormerGNN and several best performed KG and HKG embedding models on 4 HKG datasets.

Model	Cleaned JF17k [8]			WD50k [8]			Wikipeople [10]			FBAUTO [7]		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
StarE [8]	0.381	0.284	0.577	0.346	0.268	0.494	0.491	0.401	0.646	0.346	0.268	0.494
QUAD [30]	0.402	0.305	0.602	0.349	0.275	0.487	0.493	0.422	0.619	<b>0.868</b>	<b>0.836</b>	<b>0.921</b>
HAHE [26]	0.405	0.310	0.585	0.360	0.288	0.496	0.479	<u>0.430</u>	0.627	0.848	0.817	<u>0.906</u>
HyperFormer [15]	0.425	<u>0.328</u>	0.613	<u>0.372</u>	<u>0.293</u>	0.522	0.488	0.382	<u>0.652</u>	0.819	0.789	0.873
NBFNet + Prune [50]	0.415	0.313	<b>0.630</b>	0.360	0.271	<b>0.530</b>	<b>0.513</b>	<b>0.444</b>	0.644	0.767	0.700	0.888
CompGCN + Hyper [33]	0.389	0.294	0.590	0.350	0.277	0.488	0.511	0.439	0.639	0.857	0.834	0.903
<b>FormerGNN</b>	<b>0.432</b>	<b>0.335</b>	<b>0.624</b>	<b>0.377</b>	<b>0.299</b>	<u>0.527</u>	0.494	0.392	<b>0.653</b>	0.855	0.828	0.903

the reason behind these results has been widely studied in the KGE field [48, 50], on HKGs, researchers usually overlook the importance of a wide receptive field and consider the qualifiers as a substitution [15, 36, 39, 42]. In this paper, we argue that a wide receptive field is also crucial for HKGs as a complement to qualifier information, rather than a substitution. As the main-triples typically contain much more important information than qualifiers [28], the long-range dependencies formed by main-triples are also crucial.

On the other hand, although some HKGE models have wide receptive fields and better capture the graph’s long-range dependencies, as shown in Figure 3, they can suffer from information compression issues. This occurs when they aggregate qualifier information in  $h_{qe}$  and  $h_{qr}$  into the fixed-sized main-triple entity and relation embedding matrices  $h_s$  and  $h_r$ . This not only limits the models’ ability to integrate qualifiers but also introduces noise into the main-triple embedding matrices where the graph’s long-range dependencies are stored. Our claim can be justified with the following observations from Table 3:

- Firstly, although having a wide receptive field, the performance of StarE, QUAD and HAHE still falls behind HyperFormer on Cleaned JF17k and WD50k dataset.
- Secondly, the KGE model CompGCN with the hyper method can also perform comparably or even better than its HKG extension StarE and QUAD.

The reason for these results is that StarE, QUAD, and HAHE simply aggregate qualifiers into the fixed-sized entity or relation embedding matrix. This aggregation compresses the noise from qualifiers

into the main-triple embeddings. For StarE and QUAD, these embeddings were later used for GNN message passing, which even further affects the model’s ability to capture the information of the whole graph. This issue, known as over-squashing [31] in KG (Further details of over-squashing are discussed in Appendix A), can be even more pronounced in HKGs, as qualifier information can overwhelm the more important main triple information in the generated embedding matrices.

To further justify our claim, as well as propose a future research direction, we propose FormerGNN in section 5, which can be considered a qualifier integrator plus a graph encoder to capture the graph’s long-range dependency formed by main-triples, as well as the best-performed KGE NBFNet with a better approach to integrating qualifier and main-triple.

## 5 FORMERGNN

As discussed in previous sections, to support our claims and suggest a future direction for HKGE research, we need to capture the original HKG topology, achieve a wide receptive field, and improve the integration of qualifiers and main triples while avoiding information compression. To address these challenges, we propose the FormerGNN framework. FormerGNN combines a Transformer-based qualifier integrator to capture qualifier information from the original HKG, along with a GNN-based graph encoder to capture the graph’s long-range dependencies. Finally, it performs predictions by jointly using main-triple and qualifier information without aggregating them into fix-sized embedding matrixes.

- Firstly, FormerGNN's graph encoder  $GE$  would generate a graph topology embedding matrix  $h_{gt}^s$  of the subject entity  $s$  with the decomposed HKG by the prune method  $T_{prune}$ :

$$h_{gt}^s = GE(T_{prune}(\mathcal{G})). \quad (13)$$

Where  $GE$  would generate the embedding of all main-triple entities, and then apply a linear transformation to make the generated matrix align with other embedding matrixes generated by the qualifier integrator.

- Secondly, the graph topology embedding matrix would be concatenated with the main-triple embedding matrixes  $h_s, h_r, h_{[MSK]}$  and qualifier embedding matrixes  $h_{qr}, h_{qe}, \dots$ , from the qualifier integrator  $QI$  to form the decoder input  $h_{cat}$  for joint prediction:

$$h_s, h_r, h_{[MSK]}, h_{qr_1}, h_{qe_1}, \dots \leftarrow QI(F) \quad (14)$$

$$h_{cat} = Cat(h_s, h_r, h_{[MSK]}, h_{qr_1}, h_{qe_1}, \dots, h_{gt}^s). \quad (15)$$

- Lastly, this concatenation would be passed to the transformer decoder to generate the distribution  $D_{\mathcal{V}}$  over the graph node set  $\mathcal{V}$  from the embedding matrix of the mask token  $[MSK]$ , avoid any direct aggregation:

$$D_{\mathcal{V}} = Trm(h_{cat})^{[MSK]} \odot V. \quad (16)$$

As shown in Table 6, FormerGNN benefits from the advancements discussed earlier and shows improved performance compared to current HKGE models. When using NBFNet as the graph encoder and HyperFormer as the qualifier integrator, FormerGNN achieves state-of-the-art performance on both Cleaned JF17k and WD50k, highlighting the importance of the previously mentioned aspects. Note that both the qualifier integrator and the graph encoder can be substituted with more advanced transformer-based models and KG encoders. FormerGNN can easily benefit from the advances in fact or node level HKG and KGE research.

On Wikipeople, which consists mostly of triples without qualifiers, the HKG-specific module of HyperFormer struggles to train effectively due to the lack of qualifiers, resulting in sub-optimal performance compared to KGE models NBFNet and CompGCN that apply appropriate decomposition methods. However, FormerGNN still achieves comparable or better performance than all other tested HKGE models.

On FBAUTO, which only has over 8000 entities and 8 relations, the lightweight QUAD model performs best. However, FormerGNN still outperforms both HyperFormer and NBFNet on this dataset. These results further emphasize the significance of capturing HKG's original topology, capturing long-range dependencies, and integrating qualifier information without simple aggregation.

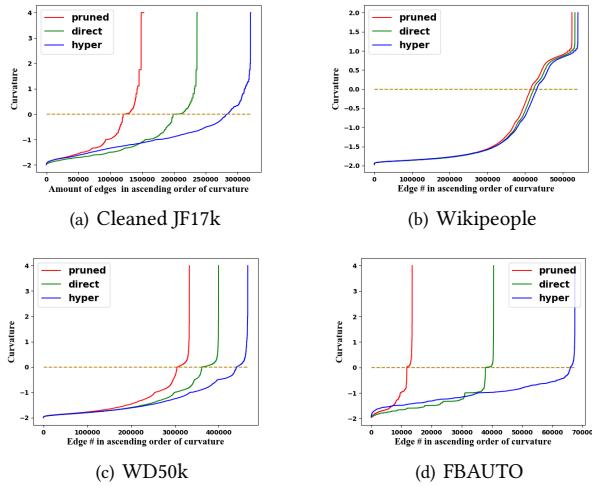
## 6 CONCLUSION

In this paper, we investigate the performance of embedding models on HKGs to determine whether HKGEs notable performance is due to the KGE base model or the specially designed qualifier processing module. To conduct this study, we convert HKGs into KG format with three decomposition methods that preserve HKG information to varying degrees.

The results show that some novel GNN-based KGE models perform comparably to classic HKGE models. However, upon further analysis, we find that HKGE models are still necessary, as the decomposition methods disrupt the original HKG topology and can

lead to information loss. Additionally, the sub-optimal performance of HKGE models can be attributed to either insufficient capture of the graph's long-range dependencies or the information compression issue caused by their aggregation of qualifier information into fix-sized main-triple embedding matrices.

To further support our claims and suggest a potential future direction for HKGE research, we propose the FormerGNN framework. This framework addresses the issues present in current KGE and HKGE models applied to HKGs. Experimental results show that FormerGNN achieves comparable or better performance than existing KGE and HKGE models on HKGs, and can easily benefit from the future advances of KGE models as the graph encoder.



**Figure 5: Curvature distribution for three decomposition methods on four datasets**

## A APPENDIX: OVER-SQUASHING

In detail, the over-squashing issue refers to the phenomenon of severe information compression that occurs in neighbor-aggregated GNNs when propagating messages from distant entities, which can lead to a loss of graph’s multi-hop topology information. It tends to occur because exponentially expanded amount of messages from distant entities need to be compressed into fixed-sized vectors when passing to the target entity [3, 31]. Over-squashing has become one of the main issues that hinder neighbor-GNNs’ performance. Since our direct and hyper decomposition methods partially introduce new nodes and edges to the graph topology and treat these newly introduced triples in the same manner as main-triples, they can further exacerbate the over-squashing issue compared to other GNN-based HKG embedding methods, such as StarE and QUAD.

To better access the over-squashing issue in a quantitative way, [31] conducts thorough analysis on graph topology and proposes the Balanced Forman curvature:

$$Ric(s, o) = \frac{2}{d_s} + \frac{2}{d_o} - 2 + 2 \frac{|\#_{\Delta}(s, o)|}{\max(d_s, d_o)} + \frac{|\#_{\Delta}(s, o)|}{\min(d_s, d_o)} + \frac{\gamma_{\max}^{-1}(s, o)}{\max(d_s, d_o)} (|\#^s_{\square}| + |\#^o_{\square}|), \quad (17)$$

where  $d_s$  is the degree of entity  $s$ ,  $\#_{\Delta}(s, o)$  are the triangles based on edge  $(s, o)$ ,  $\#^s_{\square}$  are the neighbors of entity  $o$  forming a 4-cycle based on  $(s, o)$  without diagonals inside,  $|\cdot|$  is the set size,  $\gamma_{\max}(s, o)$  is the maximal number of 4-cycles based on  $(s, o)$  traversing a common entity, and  $Ric(s, o) \in (-2, \infty)$ . Particularly, when  $Ric(s, o) < 0$ , it may refer to the occurrence of over-squashing on certain edge  $(s, o)$ .

Figure 5 displays the curvature distribution of three decomposition methods across four datasets. The x-tick represents the edges numbered in ascending order of their curvature, the y-tick represents the curvature of edges. When the curvature of an edge is less than or equal to 0, we consider it to have an over-squashing issue. Considering the prune decomposition as the baseline, the gap between other curves (the green and blue curve) and its respective curve (the red curve) can be used to assess the level of exacerbation

introduced by other decomposition methods when incorporating qualifiers in the same way as main-triples. Notably, the Cleaned JF17k and FBAUTO datasets exhibit the largest gaps, as they comprise the largest proportion of hyper-relational facts. In contrast, the Wikipeople dataset retains the smallest gap, as it contains only 2.6% of hyper-relational facts.

## GENAI USAGE DISCLOSURE

The authors only used Grammarly to improve this paper’s spelling, grammar, punctuation, clarity, and engagement. No sections or subsections in this paper are solely generated by AI.

## REFERENCES

- [1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research* 22, 82 (2021), 1–6. <http://jmlr.org/papers/v22/20-825.html>
- [2] Dimitrios Alivanistos, Max Berrendorf, Michael Cochez, and Mikhail Galkin. 2022. Query Embedding on Hyper-Relational Knowledge Graphs. In *International Conference on Learning Representations*.
- [3] Uri Alon and Eran Yahav. 2021. On the Bottleneck of Graph Neural Networks and its Practical Implications. *arXiv:2006.05205 [cs.LG]*
- [4] Heng Chang, Jiangnan Ye, Alejo Lopez-Avila, Jinhua Du, and Jia Li. 2024. Path-based explanation for knowledge graph completion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 231–242.
- [5] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
- [6] Zifeng Ding, Jingcheng Wu, Jingpei Wu, Yan Xia, Bo Xiong, and Volker Tresp. 2024. Temporal Fact Reasoning over Hyper-Relational Knowledge Graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 355–373.
- [7] Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2021. Knowledge hypergraphs: prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2191–2197.
- [8] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message Passing for Hyper-Relational Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7346–7359.
- [9] Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2020. Neuinfer: Knowledge inference on n-ary facts. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 6141–6151.
- [10] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The world wide web conference*. 583–593.
- [11] Saiping Guan, Jiayao Wei, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. Look Globally and Reason: Two-stage Path Reasoning over Sparse Knowledge Graphs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 695–705.
- [12] Richard S. Hamilton. 1986. The Ricci flow on surfaces. <https://api.semanticscholar.org/CorpusID:115319278>
- [13] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International conference on learning representations*.
- [14] Zhongnai Hou, Xiaolong Jin, Zixuan Li, Long Bai, Saiping Guan, Yutao Zeng, Jiafeng Guo, and Xueqi Cheng. 2023. Temporal knowledge graph reasoning based on n-tuple modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 1090–1100.
- [15] Zhiwei Hu, Victor Gutiérrez-Basulto, Zhiqiang Xiang, Ru Li, and Jeff Z Pan. 2023. HyperFormer: Enhancing entity and relation interaction for hyper-relational knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 803–812.
- [16] Yubo Huang and Guosun Zeng. 2024. RD-P: A Trustworthy Retrieval-Augmented Prompter with Knowledge Graphs for LLMs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 942–952.
- [17] Louis Mozart Kamdem Teyou, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2024. Embedding Knowledge Graphs in Function Spaces. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1070–1079.
- [18] Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems* 31 (2018).

- [19] Thomas N Kipf and Max Welling. 2022. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- [20] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.
- [21] Jiecheng Li, Xudong Luo, Guangquan Lu, and Shichao Zhang. 2025. Hyper-Relational Knowledge Representation Learning with Multi-Hypergraph Disentanglement. In *Proceedings of the ACM on Web Conference 2025*. 3288–3299.
- [22] Lijie Li, Hui Wang, Jiahang Li, Xiaodi Xu, Ye Wang, and Tao Ren. 2024. Integrating Structure and Text for Enhancing Hyper-relational Knowledge Graph Representation via Structure Soft Prompt Tuning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1226–1234.
- [23] Zhuofeng Li, Haoxiang Zhang, Qiannan Zhang, Ziyi Kou, and Shichao Pei. 2024. Learning from novel knowledge: Continual few-shot knowledge graph completion. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1326–1335.
- [24] Yi Liu, Hongrui Xuan, Bohan Li, Meng Wang, Tong Chen, and Hongzhi Yin. 2023. Self-supervised dynamic hypergraph recommendation based on hyper-relational knowledge graph. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1617–1626.
- [25] Yu Liu, Quanming Yao, and Yong Li. 2021. Role-aware modeling for n-ary relational knowledge bases. In *Proceedings of the Web Conference 2021*. 2660–2671.
- [26] 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. *arXiv preprint arXiv:2305.06588* (2023).
- [27] Vardaan Pahuja, Weidi Luo, Yu Gu, Cheng-Hao Tu, Hong-You Chen, Tanya Berger-Wolf, Charles Stewart, Song Gao, Wei-Lun Chao, and Yu Su. 2024. Reviving the Context: Camera Trap Species Classification as Link Prediction on Multimodal Knowledge Graphs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1825–1835.
- [28] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*. 1885–1896.
- [29] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. 2024. Graph transformers: A survey. *arXiv preprint arXiv:2407.09777* (2024).
- [30] Harry Shomer, Wei Jin, Juanhui Li, Yao Ma, and Hui Liu. 2023. Learning representations for hyper-relational knowledge graphs. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*. 253–257.
- [31] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.
- [32] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.
- [33] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.
- [34] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [35] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
- [36] Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021. Link Prediction on N-ary Relational Facts: A Graph-based Approach. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 396–407.
- [37] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.
- [38] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 1300–1307.
- [39] Bo Xiong, Mojtaba Nayyeri, Shirui Pan, and Steffen Staab. 2023. Shrinking Embeddings for Hyper-Relational Knowledge Graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 13306–13320.
- [40] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*. 1271–1279.
- [41] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*.
- [42] Donghan Yu and Yiming Yang. 2021. Improving hyper-relational knowledge graph completion. *arXiv preprint arXiv:2104.08167* (2021).
- [43] Chaohao Yuan, Kangfei Zhao, Ercan Engin Kuruoglu, Liang Wang, Tingyang Xu, Wenbing Huang, Deli Zhao, Hong Cheng, and Yu Rong. 2025. A survey of graph transformers: Architectures, theories and applications. *arXiv preprint arXiv:2502.16533* (2025).
- [44] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems* 32 (2019).
- [45] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
- [46] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 world wide web conference*. 1185–1194.
- [47] Tianli Zhang, Tongya Zheng, Zhenbang Xiao, Zulong Chen, Liangyue Li, Zunlei Feng, Dongxiang Zhang, and Mingli Song. 2024. Language Models-enhanced Semantic Topology Representation Learning For Temporal Knowledge Graph Extrapolation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3227–3236.
- [48] Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. 2023. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3446–3457.
- [49] Zhiqiang Zhang, Liqiang Wen, and Wen Zhao. 2024. A gail fine-tuned llm enhanced framework for low-resource knowledge graph question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3300–3309.
- [50] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Khonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems* 34 (2021), 29476–29490.