

# Generalizing to Unseen Elements: A Survey on Knowledge Extrapolation for Knowledge Graphs

Mingyang Chen<sup>1</sup>, Wen Zhang<sup>2</sup>, Yuxia Geng<sup>1</sup>, Zezhong Xu<sup>1</sup>, Jeff Z. Pan<sup>3</sup>, Huajun Chen<sup>1,4,5\*</sup>

<sup>1</sup>College of Computer Science and Technology, Zhejiang University

<sup>2</sup>School of Software Technology, Zhejiang University

<sup>3</sup>School of Informatics, The University of Edinburgh

<sup>4</sup>Donghai Laboratory

<sup>5</sup>Alibaba-Zhejiang University Joint Institute of Frontier Technologies

{mingyangchen, zhang.wen, gengyx, xuzezhong, huajunsir}@zju.edu.cn, j.z.pan@ed.ac.uk

## Abstract

Knowledge graphs (KGs) have become valuable knowledge resources in various applications, and knowledge graph embedding (KGE) methods have garnered increasing attention in recent years. However, conventional KGE methods still face challenges when it comes to handling unseen entities or relations during model testing. To address this issue, much effort has been devoted to various fields of KGs. In this paper, we use a set of general terminologies to unify these methods and refer to them collectively as *Knowledge Extrapolation*. We comprehensively summarize these methods, classified by our proposed taxonomy, and describe their interrelationships. Additionally, we introduce benchmarks and provide comparisons of these methods based on aspects that are not captured by the taxonomy. Finally, we suggest potential directions for future research.

## 1 Introduction

A knowledge graph (KG) [Pan *et al.*, 2017] comprises triples in the form of facts, such as (*head entity, relation, tail entity*), with entities and relations represented as nodes and edges in a graph. Despite their simplicity, KGs are increasingly recognized as valuable knowledge resources for various applications.

Knowledge graph embedding (KGE) is a dominant field that has recently gained massive attention for representing elements (i.e., entities and relations) of knowledge graphs in latent vector spaces. Conventional KGE methods are evaluated among entities and relations occurring in triples used for model training, assuming a fixed entity and relation set. However, this assumption does not always hold in real-world applications [Hamaguchi *et al.*, 2017; Xiong *et al.*, 2018].

Knowledge graphs are dynamic, with new entities and relations emerging over time. Furthermore, the number of knowledge graphs is increasing. Conventional KGE methods are incapable of handling emerging elements of KGs because they

cannot map unseen elements to a proper position in the vector space of trained elements. This inability to generalize to unseen elements limits their usefulness in addressing the evolution feature of KGs.

Recently, many studies have focused on generalizing to unseen elements of KGs in various scenarios. For instance, some research has concentrated on predicting missing triples for out-of-knowledge-base (*OOKB*) entities [Hamaguchi *et al.*, 2017; Wang *et al.*, 2019]. Additionally, some *inductive* relation prediction methods and toolkits have studied generalizing to entirely new KGs with unseen entities [Teru *et al.*, 2020; Chen *et al.*, 2022b; Zhang *et al.*, 2023]. Furthermore, problems of generalizing to unseen relations have also been deeply investigated, especially in low-resource settings such as *few-shot* [Xiong *et al.*, 2018; Chen *et al.*, 2019] and *zero-shot* [Geng *et al.*, 2021] scenarios.

Therefore, the problem of generalizing to unseen elements has garnered increasing attention in the knowledge graph field. However, investigations of this problem have used different scenarios and terminologies, such as *OOKB*, *inductive*, *few-shot*, and *zero-shot*. Furthermore, there is no existing literature that systematically summarizes this area. Based on our research in this field, we ask the following question: *Can we use a general set of terminologies to comprehensively summarize works in this area and make comparisons to provide insights for future research?* In this paper, we present the first survey of recent studies on generalizing to unseen elements in KGs and refer to them collectively as *Knowledge Extrapolation*. Specifically, we begin by providing background on knowledge extrapolation and unifying related definitions. Next, we offer an overview of knowledge extrapolation methods and categorize them using our proposed taxonomy. Additionally, we summarize commonly used benchmarks, compare methods, and suggest directions for future research.

## 2 Preliminary

### 2.1 Knowledge Graph Embedding

To begin, we provide a formal definition of knowledge graphs. Knowledge graphs are commonly represented as graph structures, in which nodes represent entities corresponding to specific concepts, and edges are labeled by relations. Formally, a knowledge graph can be denoted as

\*Corresponding author.

$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , where  $\mathcal{E}$  is the set of entities,  $\mathcal{R}$  is the set of relations, and  $\mathcal{T}$  is the set of triples. A triple describes a connection between two entities through a relation, and can be represented as  $(h, r, t) \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ . In this context, both entities and relations are considered elements within a KG, as is common in the RDF [Pan, 2009] format.

Based on knowledge graphs, the goal of conventional knowledge graph embedding methods is to embed elements in  $\mathcal{E}$  and  $\mathcal{R}$  into continuous vector spaces while preserving the inherent structure of a KG [Wang *et al.*, 2017]. Link prediction is a common task for evaluating the effectiveness of a KGE method, and we use it as an example to introduce the basics since most existing works focus on it. However, exploring KGE and knowledge extrapolation on other KG-related tasks is a pressing need (cf. §6.1).

In practice, as shown in Figure 1, there is a training set  $\mathcal{G}^{tr} = (\mathcal{E}^{tr}, \mathcal{R}^{tr}, \mathcal{T}^{tr})$  (a) and a test set  $\mathcal{G}^{te} = (\mathcal{E}^{te}, \mathcal{R}^{te}, \mathcal{T}^{te})$  (b). The goal of link prediction is to train the model to score positive triples higher than negative triples and generalize the model to the test set. This can be expressed as  $\min_{\theta} \mathbb{E}_{(h,r,t) \in \mathcal{T}^{te}} [\ell_{\theta}(h, r, t)]$ , where  $\ell_{\theta}(h, r, t) \propto -s(\mathbf{h}, \mathbf{r}, \mathbf{t}) + s(\mathbf{h}', \mathbf{r}', \mathbf{t}')$  is the loss function,  $s$  is the score function, and  $\theta$  represents the learnable parameters, including entity and relation embeddings. We use  $\mathbf{h}, \mathbf{r}, \mathbf{t}$  to denote the embeddings of  $h, r, t$  and  $(\mathbf{h}', \mathbf{r}', \mathbf{t}') \notin \mathcal{T}^{tr}$  to denote negative samples.

However, conventional KGE methods cannot deal with new entities and relations during the test since they don't have proper embeddings for new elements. Knowledge extrapolation focuses on solving this problem, and we describe the details next.

## 2.2 Knowledge Extrapolation Settings

Methods aimed at knowledge extrapolation attempt to conduct link prediction on unseen elements. To unify existing works on handling these unseen elements, we introduce a set of general terms. Specifically, during knowledge extrapolation, there are two sets used for testing: one provides support information about the unseen elements (such as their structural or textual features), and the other evaluates the model's link prediction ability, much like the original  $\mathcal{T}^{te}$ . We refer to these sets as the support set  $\mathcal{S}^{te}$  and query set  $\mathcal{Q}^{te}$ , respectively, and the test set is formulated as  $\mathcal{G}^{te} = (\mathcal{E}^{te}, \mathcal{R}^{te}, \mathcal{S}^{te}, \mathcal{Q}^{te})$ . While different works may use varying terminology, they must all involve these two sets during knowledge extrapolation. For convenience, we refer to them uniformly as the support and query sets.

In this work, we categorize existing approaches for handling unseen elements into two types: *Entity Extrapolation* and *Relation Extrapolation*. As illustrated in Figure 1, we use the term "entity extrapolation" to refer to situations where previously unseen entities appear in the test set, and "relation extrapolation" to describe scenarios where previously unseen relations are present in the test set.

Recent years have seen fast growth in knowledge extrapolation for KGs. Early work on rule learning [Galárraga *et al.*, 2013; Meilicke *et al.*, 2018] addressed entity extrapolation via rules that are independent of specific entities. Later, researchers [Xie *et al.*, 2016; Shah *et al.*, 2019] used tex-

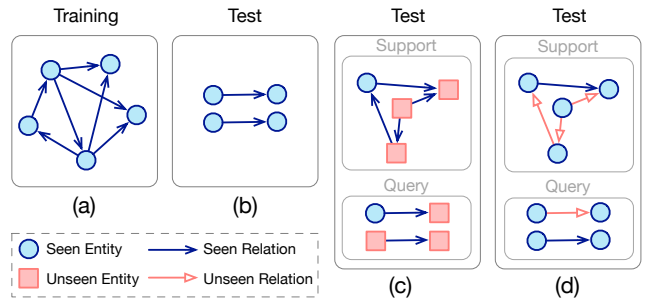


Figure 1: Training (a) and test (b) set for conventional knowledge graph embedding. Example test set for the entity extrapolation setting (c) and the relation extrapolation setting (d). Note that there may be any support information about unseen elements in the support set, and we use relevant triples as examples.

tual descriptions to handle unseen entities. With GNNs becoming popular for KGs, methods [Hamaguchi *et al.*, 2017; Wang *et al.*, 2019] investigated aggregating knowledge from seen entities to unseen entities. GraIL [Teru *et al.*, 2020] used subgraph encoding to handle unseen entities in a completely new KG, with many methods following this paradigm. Other works [Chen *et al.*, 2022b; Galkin *et al.*, 2022a] explored encoding entity-independent information for unseen entities. Meanwhile, some models [Wang *et al.*, 2021c; Wang *et al.*, 2021a] incorporated pre-trained language models to encode textual descriptions for unseen entities. Recent research has also considered few-shot settings for handling unseen entities [Baek *et al.*, 2020; Zhang *et al.*, 2021]. For relation extrapolation, pioneer works [Xiong *et al.*, 2018; Chen *et al.*, 2019] have often focused on few-shot settings. Other works represent unseen relations using textual descriptions [Qin *et al.*, 2020] or ontologies [Geng *et al.*, 2021], without using support triples, referred to as zero-shot learning for KGs.

## 3 Methods

In this section, we describe the existing works on knowledge extrapolation in detail. As shown in Figure 2, we categorize these methods based on their model design. For each category of methods, we begin by introducing its general idea and then delve into the specifics of existing methods.

### 3.1 Entity Extrapolation

#### Entity Encoding-based Entity Extrapolation

Conventional knowledge graph embedding methods typically learn an embedding lookup table for entities. However, this paradigm hinders models from extrapolating to unseen entities, as there are no reasonable embeddings for them in the learned table. To handle unseen entities, an intuitive approach is to learn to encode entities rather than learn fixed embedding tables. These learned encoders can operate on the support set of entities to produce reasonable embeddings for them. The score function of entity encoding-based entity extrapolation for  $(h, r, t) \in \mathcal{Q}^{te}$  is formulated as:

$$s(f(h, \mathcal{S}^{te}), \mathbf{r}, f(t, \mathcal{S}^{te})) \quad (1)$$

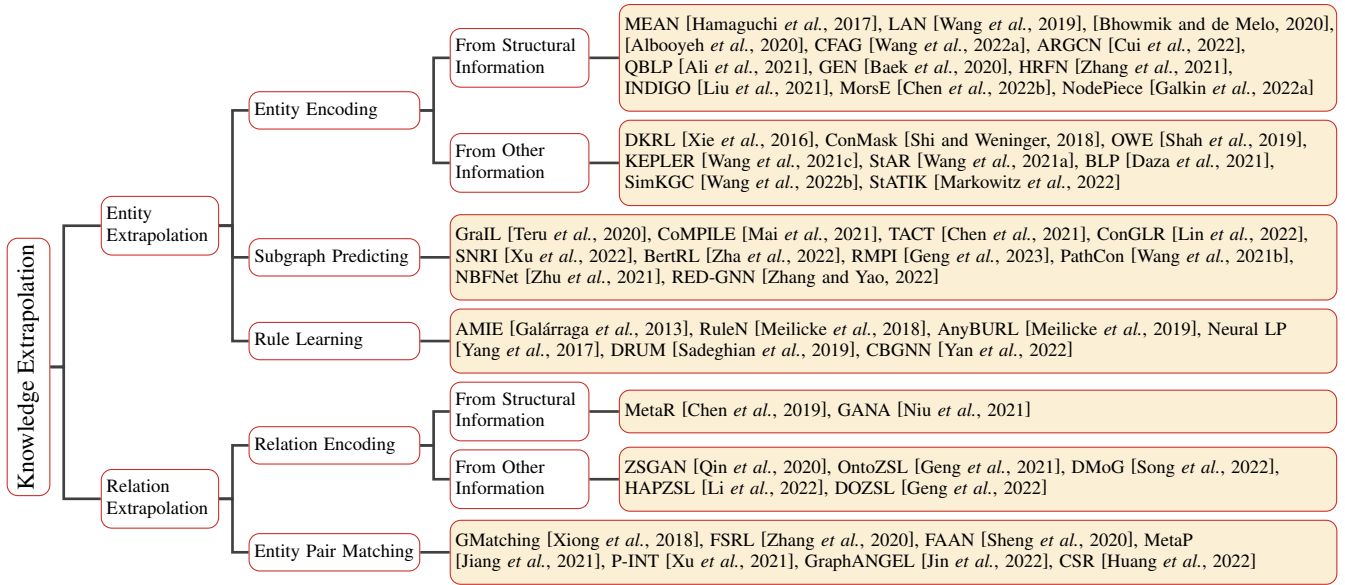


Figure 2: Taxonomy of knowledge extrapolation for knowledge graphs.

where  $f$  denotes the learnable encoder, and  $r$  denotes relation embedding or relation-dependent parameters. Note that not all methods encode both head and tail entities.

Existing works have designed various encoding models  $f$  that correspond to different types of information in the support set  $S^{te}$ . If the support set contains triples about unseen entities, then  $f$  encodes those entities *from structural information*. On the other hand, if the support set contains other types of information (e.g., textual descriptions) about unseen entities, we refer to this situation as encoding unseen entities *from other information*.

**Encode from structural information.** Some methods assume that unseen entities are connected to seen entities and focus on transferring knowledge from the latter to the former by aggregating neighbors for the unseen entities. For example, Hamaguchi *et al.* [2017] apply a graph neural network to propagate representations between entities and represent unseen emerging entities using different transition and pooling functions. To build more efficient neighbor aggregators for emerging entities, LAN [Wang *et al.*, 2019] uses attention weights based on logic rules and neighbor entity embeddings that consider query relations. VN Network [He *et al.*, 2020] conducts virtual neighbor prediction based on logic and symmetric path rules to make the neighbor for an unseen entity denser. Bhowmik and de Melo [2020] propose a graph transformer architecture to aggregate the neighbor information for learning entity embeddings and use the policy gradient to find a symbolic reasoning path for explainable reasoning. Albooyeh *et al.* [2020] designs a training procedure resembling what is expected of the model at the test time. CFAG [Wang *et al.*, 2022a] first uses an aggregator to obtain entity representations and then applies a GAN to generate entity representations based on the query relation. ARGCN [Cui *et al.*, 2022] considers representing new entities emerging in multiple batches. QBLP [Ali *et al.*, 2021] explores the benefits of

employing hyper-relational KGs on entity extrapolation.

In addition to considering connections between seen and unseen entities, recent works have also started to explore relations between unseen entities. GEN [Baek *et al.*, 2020] considers unseen entities with only a few associative triples and tackles link prediction between seen to unseen entities, as well as between unseen entities, by designing a meta-learning framework to conduct learning to extrapolate the knowledge from a given KG to unseen entities. Following GEN, HRFN [Zhang *et al.*, 2021] first coarsely represents unseen entities by meta-learned hyper-relation features and then uses a GNN to obtain fine-grained embeddings for unseen entities.

In certain scenarios, unseen entities have no connection with seen entities and form a completely new KG. Consequently, some methods assume that the training and test entity sets are not overlapping, i.e.,  $\mathcal{E}^{te} \cap \mathcal{E}^{tr} = \emptyset$ . For these scenarios, the methods do not focus on aggregating representations from seen entities for unseen entities. Instead, they learn to encode entity-independent information to output entity embeddings. MorsE [Chen *et al.*, 2022b] uses an entity initializer and a GNN modulator to model entity-independent meta-knowledge to produce high-quality embeddings for unseen entities and learn such meta-knowledge by meta-learning. NodePiece [Galkin *et al.*, 2022a] also treats connected relations as entity-independent information and tries MLP and Transformer to encode relational contexts for unseen entities. Unlike the above methods encoding entities individually, INDIGO [Liu *et al.*, 2021] encodes the support triples and query candidate triples into a node-annotated graph where nodes correspond to pairs of entities, and uses GNN to update node features for indicating the query triple predictions.

**Encode from other information.** In addition to using relevant triples as support sets for unseen entities, other types of information can also be useful for encoding unseen entities. The textual description is common for encoding un-

seen entities. DKRL [Xie *et al.*, 2016] embodies entity descriptions for knowledge graph embedding and jointly trains structure-based embeddings and description-based embeddings. ConMask [Shi and Weninger, 2018] applies relationship-dependent content masking on a given entity description and uses the representation extracted from the description to predict the most proper entities from an existing entity set. OWE [Shah *et al.*, 2019] trains traditional knowledge graph embeddings and word embeddings independently, and learns a transformation from entities’ textual information to their embeddings, enabling unseen entities with text can be mapped into the KGE space. KEPLER [Wang *et al.*, 2021c] uses the same pre-trained language model to encode entities and texts, and jointly optimizes it with knowledge embedding and masked language modeling objectives. StAR [Wang *et al.*, 2021a] is a structure-augmented text representation model which encodes triples with a siamese-style textual encoder. Since it encodes the text of entities, it can easily extrapolate to unseen entities using their text. BLP [Daza *et al.*, 2021] treats a pre-trained language model as an entity encoder and finetunes it via link prediction. SimKGC [Wang *et al.*, 2022b] uses contrastive learning with different negative samplings to improve the efficiency of text-based knowledge graph embedding. STATIK [Markowitz *et al.*, 2022] combines a language model and a message passing graph neural network as an encoder to process and uses TransE as the decoder to score triples.

### Subgraph Predicting-based Entity Extrapolation

The aforementioned entity encoding-based methods commonly treat the head entity, relation, and tail entity in a triple individually. However, some research provides another view that does not explicitly encode entities. Instead, they treat the head and tail entity in a triple together and encode the relational subgraph between them. This perspective assumes that the semantics underlying the subgraph between two entities can be used to predict their relation. The ability to encode the subgraph of two entities can be extrapolated to unseen entities since subgraph structures are independent of entities. For the subgraph predicting-based method, the score function can be formulated as:

$$s(f(h, t, \mathcal{S}^{te}), \mathbf{r}) \quad (2)$$

where  $f$  is responsible for subgraph extraction and encoding;  $\mathbf{r}$  denotes relation embedding or relation-dependent parameters;  $s$  is responsible for predicting whether the subgraph is related to a specific relation. This research line mainly encodes subgraphs for unseen entities based on triples from support sets, so they assume that support sets contain structural information about unseen entities.

One line of this method starts from GraIL [Teru *et al.*, 2020], which uses a GNN to reason over enclosing subgraph between target nodes and learns entity-independent relational semantics for entity extrapolation. Specifically, for predicting the relation between two unseen entities, GraIL first extracts the enclosing subgraph around them, then labels each entity in this subgraph based on the distance between that entity and two unseen entities. Finally, it uses a GNN to score the subgraph with candidate relations. CoMPLE [Mai *et al.*,

2021] proposes a node-edge communicative message-passing mechanism replacing the original GNN to consider the importance of relations. TACT [Chen *et al.*, 2021] considers correlations between relations in a subgraph, and encodes a Relational Correlation Network (RCN) to enhance the encoding of the enclosing subgraph. ConGLR [Lin *et al.*, 2022] formulates a context graph representing relational paths from a subgraph and then uses two GCNs to process it with the enclosing subgraph respectively. Since extracted enclosing subgraphs can be sparse and some surrounded relations are neglected, SNRI [Xu *et al.*, 2022] leverages complete neighbor relations of entities in a subgraph by neighboring relational features and neighboring relational paths. BertRL [Zha *et al.*, 2022] leverages pre-trained language models to encode reasoning paths to enhance subgraph semantics. RMPI [Geng *et al.*, 2023] utilizes the relation semantics defined in the KG’s ontological schema [Wiharja *et al.*, 2020] to extend for handling both unseen entities and relations.

The path between two entities can also be viewed as a subgraph and used to indicate the relation between them. Path-Con [Wang *et al.*, 2021b] proposes a relational message passing framework and conducts it to encode relational contexts and relational paths for two given entities to predict the missing relation. NBFNet [Zhu *et al.*, 2021] uses the Message and Aggregate operation from GNN to parameterize the generalized Bellman-Ford algorithm for representing the paths between two entities. RED-GNN [Zhang and Yao, 2022] proposes a relational directed graph (r-digraph) to preserve the local information between entities and recursively constructs the r-digraph between one query and any candidate entity, making the reasoning processes efficient.

### Rule Learning-based Entity Extrapolation

Several works have explored learning rules from knowledge graphs, as these logical rules can inherently extrapolate to unseen entities since they are independent of specific entities. Rule learning-based methods can be divided into two categories. The pure symbolic methods generate rules from existing knowledge through statistics and filter them with predefined indicators. AMIE [Galárraga *et al.*, 2013] proposes predefined indicators, including head coverage and confidence, to filter possible rules which are generated with three different atom operators. RuleN [Meilicke *et al.*, 2018] and AnyBURL [Meilicke *et al.*, 2019] generate possible rules by sampling the path from the head entity to the tail entity of each triple and filtering them using confidence and head coverage. Another kind of method combines neural networks and symbolic rules. Neural LP [Yang *et al.*, 2017] and DRUM [Sadeghian *et al.*, 2019] are based on TensorLog and use neural networks to simulate rule paths. CBGNN [Yan *et al.*, 2022] view logical rules as cycles from the perspective of algebraic topology and learn rules in the space of cycles.

## 3.2 Relation Extrapolation

### Relation Encoding-based Relation Extrapolation

Similar to entity extrapolation, the shortcoming of conventional knowledge graph embedding methods on relation extrapolation is that they cannot give unseen relations reasonable embeddings. However, since some observed information

from the support set for unseen relations can be utilized, encoding such information to embed unseen relations is an intuitive solution. The score function of relation encoding-based methods is formulated as follows:

$$s(\mathbf{h}, f(r, \mathcal{S}^{te}), \mathbf{t}) \quad (3)$$

where  $f$  is an encoder that transforms related support information for an unseen relation  $r$  to its embedding, and  $\mathbf{h}$  and  $\mathbf{t}$  are entity embeddings or entity-dependent parameters. Depending on the type of information used for relation encoding, we also categorize these methods into *encode from structural information* and *encode from other information*.

**Encode from structural information.** Methods in this category assume that there are some example triples (i.e., structural information) for unseen relations in the support set, and the connected head and tail entities of unseen relations can reveal their semantics. These methods mainly focus on transferring representation from entities to unseen relations. MetaR [Chen *et al.*, 2019] learns relation-specific meta information from support triples to replace the missing representations for unseen relations. More precisely, for a specific unseen relation, MetaR first encodes entity pairs of it and outputs relation meta. Then, a rapid gradient descent step is applied to the relation meta following optimization-based meta-learning, which will produce good generalization performance on that unseen relation. In order to leverage more structural information provided by connected entities of unseen relations, GANA [Niu *et al.*, 2021] uses a gated and attentive neighbor aggregator to represent entity representations. Such entity representations are used to generate general representations for unseen relations via an attentive Bi-LSTM encoder. Furthermore, it applies the training paradigm of optimization-based meta-learning with TransH as the score function.

**Encode from other information.** In addition to using structural information from relevant triples, methods in this category rely on unseen relations’ support information from external resources, including textual descriptions and ontological schemas. ZSGAN [Qin *et al.*, 2020] leverages Generative Adversarial Networks (GANs) to generate plausible relation embeddings for unseen relations conditioned on their encoded textual descriptions. Since a KG is often accompanied by an ontology as its schema, OntoZSL [Geng *et al.*, 2021] is proposed to synthesize the unseen relation embeddings conditioned on the embeddings of an ontological schema which includes the semantics of the KG relations. DOZSL [Geng *et al.*, 2022] proposes a property-guided disentangled ontology embedding method to extract more fine-grained inter-class relationships between seen and unseen relations in the ontology. It uses a GAN-based generative model and a GCN-based propagation model to integrate disentangled embeddings for generating embeddings for unseen relations. Besides leveraging generative models such as GANs to obtain plausible relation embeddings from support information, DMOG [Song *et al.*, 2022] learns a linear mapping matrix to transform the encoded textual and ontological features, and HAPZSL [Li *et al.*, 2022] uses an attention mechanism to encode relation descriptions into valid relation prototype representations.

## Entity Pair Matching-based Relation Extrapolation

Another solution, instead of encoding the relation directly, is to encode head and tail entity pairs of an unseen relation and then match these encoded entity pairs with entity pairs in the query set to predict whether they are connected by the same unseen relation. The score function of entity pair matching-based methods for  $(h, r, t)$  can be formulated as follows:

$$s(f(\text{PAIR}(r, \mathcal{S}^{te})), f(h, t)) \quad (4)$$

$$\text{PAIR}(r, \mathcal{S}^{te}) = \{(h_i, t_i) | (h_i, r, t_i) \in \mathcal{S}^{te}\}$$

where  $\text{PAIR}(r, \mathcal{S}^{te})$  is used to get the head-tail entity pairs from the support set for the unseen relation.  $f$  encodes head-tail entity pairs, and  $s$  scores for entity pair matching. Since this research line involves encoding entity pairs for unseen relations, the support sets are commonly assumed to consist of relevant triples for these unseen relations.

GMatching [Xiong *et al.*, 2018] uses entity embeddings and entity local graph structures to represent entity pairs and learn a metric model to discover more similar triples. FSRL [Zhang *et al.*, 2020] uses a relation-aware heterogeneous neighbor encoder considering different impacts of neighbors of an entity, and a recurrent autoencoder aggregation network to aggregate multiple reference entity pairs in the support set. FAAN [Sheng *et al.*, 2020] uses an adaptive neighbor encoder to encode entities and uses a Transformer to encode entity pairs; furthermore, an adaptive matching processor is introduced to match a query entity pair to multiple support entity pairs attentively. MetaP [Jiang *et al.*, 2021] learns relation-specific meta patterns from entity pairs of a relation, and such patterns are captured by convolutional filters on representations (by pre-trained or random initialize) of entity pairs. Furthermore, the subgraphs between the entity pairs are also expressive. P-INT [Xu *et al.*, 2021] leverages the paths between an entity pair to represent it and calculates the path similarity between support entity pairs and query entity pairs to predict the query triples about unseen relations. GraphANGEL [Jin *et al.*, 2022] extracts 3-cycle and 4-cycle patterns to represent an unseen relation; specifically, for a query triple with an unseen relation, it extracts target patterns, supporting patterns, refuting patterns for query entity pairs and calculates the similarity between target patterns and support patterns as well refuting patterns. CSR [Huang *et al.*, 2022] uses connection subgraphs to represent entity pairs. It finds the shared connection subgraph among the support triples and tests whether it connects query triples.

## 4 Benchmarks

In this section, we briefly describe some datasets used to evaluate models for knowledge extrapolation. The following four datasets are commonly used by different knowledge extrapolation models, and they have various assumptions for diverse application scenarios. We show the usage details in Table 1 and describe them as follows.

① **WN11- $\{\text{Head/Tail/Both}\}$ - $\{1,000/3,000/5,000\}$**  Here are nine datasets derived from WordNet11 created by Hamaguchi *et al.* [2017]. These datasets are used to conduct entity extrapolation experiments, and they assume that

Dataset	Type	Proposed by	Used by
①	Ent	MEAN [Hamaguchi <i>et al.</i> , 2017]	MEAN [Hamaguchi <i>et al.</i> , 2017], LAN [Wang <i>et al.</i> , 2019], VN Network [He <i>et al.</i> , 2020], INDIGO [Liu <i>et al.</i> , 2021]
②	Ent	GraIL [Teru <i>et al.</i> , 2020]	GraIL [Teru <i>et al.</i> , 2020], CoMPiLE [Mai <i>et al.</i> , 2021], TACT [Chen <i>et al.</i> , 2021], ConGLR [Lin <i>et al.</i> , 2022], SNRI [Xu <i>et al.</i> , 2022], BertRL [Zha <i>et al.</i> , 2022], RMPI [Geng <i>et al.</i> , 2023], MorsE [Chen <i>et al.</i> , 2022b], NodePiece [Galkin <i>et al.</i> , 2022a], NBFNet [Zhu <i>et al.</i> , 2021], RED-GNN [Zhang and Yao, 2022], INDIGO [Liu <i>et al.</i> , 2021], RuleN [Meilicke <i>et al.</i> , 2018], Nerual LP [Yang <i>et al.</i> , 2017], DRUM [Sadeghian <i>et al.</i> , 2019], CBGNN [Yan <i>et al.</i> , 2022]
③	Rel	GMatching [Xiong <i>et al.</i> , 2018]	GMatching [Xiong <i>et al.</i> , 2018], MetaR [Chen <i>et al.</i> , 2019], FSRL [Zhang <i>et al.</i> , 2020], FAAN [Sheng <i>et al.</i> , 2020], GANA [Niu <i>et al.</i> , 2021], MetaP [Jiang <i>et al.</i> , 2021], P-INT [Xu <i>et al.</i> , 2021], CSR [Huang <i>et al.</i> , 2022]
④	Rel	ZSGAN [Qin <i>et al.</i> , 2020]	ZSGAN [Qin <i>et al.</i> , 2020], OntoZSL [Geng <i>et al.</i> , 2021], DOZSL [Geng <i>et al.</i> , 2022] DMoG [Song <i>et al.</i> , 2022], HAPZSL [Li <i>et al.</i> , 2022]

Table 1: The usage of four commonly used benchmarks described in §4. The column ‘Type’ indicates the dataset is used for entity or relation extrapolation.

unseen entities are all connected to training entities since triples that contain two unseen entities in support sets are discarded. Head/Tail/Both denotes the position of unseen entities in test triples, and 1,000/3,000/5,000 represents the number of triplets used for generating the unseen entities.

② **{WN18RR/FB15k-237/NELL995}-{v1/2/3/4}** These 12 datasets are generated by Teru *et al.* [2020] from WN18RR, FB15k-237, and NELL955. Each original dataset is sampled into four different versions with an increasing number of entities and relations. In these datasets, the entity sets during training and testing are disjoint, which indicates that they are used to evaluate entity extrapolation models assuming unseen entities are not connected to training entities.

③ **NELL-One/Wiki-One** These two datasets are developed by Xiong *et al.* [2018] and used to evaluate the few-shot relation prediction task originally. The relation sets for the training and the testing are disjoint, indicating models evaluated on these datasets are capable of relation extrapolation. For each unseen relation during the test, the number of support triples (i.e.,  $k$ ) is often specified, and the task is called  $k$ -shot link prediction.

④ **NELL-ZS/Wiki-ZS** These two datasets are presented by Qin *et al.* [2020], and ‘ZS’ denotes that they are used to evaluate zero-shot learning for link prediction, i.e., relation extrapolation with information other than triples related to unseen relations. Each relation in these datasets has its textual description, and such information is viewed as support sets for unseen relations. Furthermore, besides textual information, Geng *et al.* [2021] adds ontological schemas for these two datasets as side information for relations.

Moreover, there are many other datasets provided by different works to show their own effectiveness from some specific aspects. LAN [Wang *et al.*, 2019] and VN Network [He *et al.*, 2020] use FB15k and YAGO37 to construct datasets, respectively. GEN [Baek *et al.*, 2020] and HRFN [Zhang *et al.*, 2021] develop datasets to demonstrate their link prediction ability on both seen and unseen entities. MaKER [Chen *et al.*, 2022a] and RMPI [Geng *et al.*, 2023] construct datasets showing that they can handle both unseen entities and re-

lations during the model test. ARGCN [Cui *et al.*, 2022] and LKGE [Cui *et al.*, 2023] create datasets to simulate the growth of knowledge graphs with emerging entities and relations.

## 5 Comparison and Discussion

### 5.1 Assumption on Entity Extrapolation

As described in §3.1, there are two different assumptions about entity extrapolation. One assumption is that unseen entities in support sets are connected to seen entities (i.e.,  $\mathcal{E}^{te} \cap \mathcal{E}^{tr} \neq \emptyset$ ), while the other is that unseen entities form entirely new KGs in support sets and are not connected by seen entities (i.e.,  $\mathcal{E}^{te} \cap \mathcal{E}^{tr} = \emptyset$ ). We refer to these two assumptions as semi-entity extrapolation and fully-entity extrapolation for convenience. Generally, methods designed for fully-entity extrapolation can handle semi-entity extrapolation problems, but not vice versa. We discuss the ability of entity extrapolation methods to handle these two assumptions as follows.

Most semi-entity extrapolation models lie in entity encoding-based methods and encode unseen entities from structural information since they often design modules for transferring knowledge from seen entities to unseen entities by aggregating representations from seen entities [Hamaguchi *et al.*, 2017; Wang *et al.*, 2019; Wang *et al.*, 2022a]. Some other methods that conduct entity encoding from structural information can also handle fully-entity extrapolation by designing entity-independent encoding procedure [Chen *et al.*, 2022b; Galkin *et al.*, 2022a].

Furthermore, methods encoding unseen entities from other information (e.g., textual descriptions) can usually solve fully-entity extrapolation problems since such side information is sufficient for encoding entity embeddings and such encoding is independent to specific entities [Wang *et al.*, 2021a; Wang *et al.*, 2022b]. Subgraph predicting-based methods and rule learning-based methods are also capable of tackling fully-entity extrapolation since subgraphs and rules are entity-independent.

## 5.2 Information in Support Set

Various types of information have been explored to build support sets for unseen elements, including triples, textual descriptions, and ontologies. Here, we compare these three widely used types of support information as follows.

Triples, which provide structural information, are an intuitive type of support information for unseen elements since they usually emerge with other elements in the form of triples rather than alone. Knowledge from seen elements provided by triples can be utilized by unseen elements. For example, in entity extrapolation, surrounding seen relations can provide type information for unseen entities [Chen *et al.*, 2022b]; in relation extrapolation, connected seen entities can reveal the characteristics of unseen relations [Xiong *et al.*, 2018; Chen *et al.*, 2019].”

Textual descriptions are also common for KGs since many KGs are constructed from text data. Text descriptions can naturally provide the ability to extrapolate to unseen elements, and the typical procedure is to use text encoders (e.g., pre-trained language models) to transform text descriptions into embeddings. That is, the encoder  $f$  in Eq. (1) or Eq. (3) can be treated as a text encoder. Both entity extrapolation [Wang *et al.*, 2021a; Daza *et al.*, 2021] and relation extrapolation [Qin *et al.*, 2020] can benefit from textual descriptions.

Ontologies are typically used as prior knowledge about the correlation between seen and unseen elements and have been widely used to handle unseen relations in existing works. An ontology is often represented as a graph encompassing relation hierarchies and constraints on relation domains and ranges. The embeddings of unseen relations can be generated using an ontology-based approach that utilizes various techniques, such as GANs [Geng *et al.*, 2021] or disentangled representation learning [Geng *et al.*, 2022].

Besides using the above support information alone, some methods apply different support information together. For example, BertRL [Zha *et al.*, 2022] uses both subgraph structures and textualized paths to predict relations between unseen entities, exploiting both structural triples and textual descriptions. Similarly, RMPI [Geng *et al.*, 2023] uses subgraphs and ontological schemas together to tackle both unseen entities and relations.

## 6 Future Prospects

### 6.1 Diverse Application

Most existing knowledge extrapolation methods are evaluated by simple link prediction on test sets. Even though the link prediction task can show the effectiveness of models and help knowledge graph completion, it is valuable to explore how to generalize to unseen KG elements on diverse applications. GNN-QE [Galkin *et al.*, 2022b] studies answering logical queries expressed in a subset of first-order logic with unseen entities at test time. Meanwhile, ContEA [Wang *et al.*, 2022c] targets the entity alignment task under the growing of KGs. We argue that besides the above in-KG applications, more common out-of-KG tasks like KG enhanced question answering [Chen *et al.*, 2022b], including those based on LLMs [Hu *et al.*, 2023] can be explored.

### 6.2 Multi-modal Support Information

Multi-modal knowledge graphs are a current research topic that has been explored in recent literature. While existing knowledge extrapolation methods have primarily focused on using natural language as support information for unseen elements, there are few works that address the potential of utilizing visual information. However, we believe that images can also help generalize KGE to unseen elements since they can be universally understood by specific pre-trained encoders. Additionally, hyper-relational KGs [Ali *et al.*, 2021], where triples can be instantiated with a set of qualifiers, can provide different modal information.

### 6.3 Entity and Relation Extrapolation

Existing research on knowledge extrapolation primarily focuses on solving entity extrapolation and relation extrapolation separately, but in real-world applications, unseen entities and relations may emerge simultaneously. One feasible solution is effectively integrating methods of entity extrapolation and relation extrapolation. Existing literature has made some attempts. Specifically, MaKER [Chen *et al.*, 2022a] uses meta-learning to learn to encode unseen entities and relations based on a set of training tasks with simulated unseen entities and relations. Furthermore, by combining the capability of entity extrapolation from subgraph encoding and relation extrapolation based on the ontology of relations, RMPI [Geng *et al.*, 2023] provides a nascent exemplar of this perspective.

### 6.4 Temporal and Lifelong Setting

In practical applications, some KGs include temporal constraints, which necessitates the consideration of temporal information when scoring a triple [Bourgaux *et al.*, 2021; Liang *et al.*, 2022]. Temporal KGs also face the challenge of emerging elements due to their dynamic nature. To address this issue, FILT [Ding *et al.*, 2022] defines a problem of entity extrapolation in temporal KGs and utilizes a time-aware graph encoder and entity concept information to obtain embeddings for unseen entities. Additionally, existing knowledge extrapolation works typically assume one-time extrapolation, where all unseen elements emerge simultaneously in a batch. However, recent literature, such as ARGCN [Cui *et al.*, 2022] and LKGE [Cui *et al.*, 2023], considers scenarios where unseen elements emerge in a multi-batch and lifelong manner.

## 7 Conclusion

Recent years have seen an increasing amount of research on generalizing to unseen elements in KGs from diverse perspectives. In this paper, we provide a comprehensive survey of these works and summarize them under a general set of terminologies. We categorize existing methods using our proposed systematic taxonomy and list commonly used benchmarks along with the methods that employ them. We also provide method comparisons and discussions from commonly mentioned perspectives in the existing literature. Finally, we suggest several potential research directions. We believe that this exploration can provide a clear overview of the field and facilitate future research.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFCU19B2027, NSFC91846204), the joint project DH-2022ZY0012 from Donghai Lab and the Chang Jiang Scholars Program (J2019032). Mingyang Chen is supported by the China Scholarship Council (No. 202206320309).

## Contribution Statement

Mingyang Chen and Wen Zhang contributed equally and share first authorship.

## References

- [Albooyeh *et al.*, 2020] M. Albooyeh, R. Goel, and S. M. Kazemi. Out-of-sample representation learning for knowledge graphs. In *EMNLP*, 2020.
- [Ali *et al.*, 2021] M. Ali, M. Berrendorf, M. Galkin, V. Thost, T. Ma, V. Tresp, and J. Lehmann. Improving inductive link prediction using hyper-relational facts. In *ISWC*, 2021.
- [Baek *et al.*, 2020] J. Baek, D. Bok Lee, and S. Ju Hwang. Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. In *NeurIPS*, 2020.
- [Bhowmik and de Melo, 2020] R. Bhowmik and G. de Melo. Explainable link prediction for emerging entities in knowledge graphs. In *ISWC*, 2020.
- [Bourgaux *et al.*, 2021] C. Bourgaux, A. Ozaki, and J. Z. Pan. Geometric models for (temporally) attributed description logs. In *DL*, 2021.
- [Chen *et al.*, 2019] M. Chen, W. Zhang, W. Zhang, Q. Chen, and H. Chen. Meta relational learning for few-shot link prediction in knowledge graphs. In *EMNLP*, 2019.
- [Chen *et al.*, 2021] J. Chen, H. He, F. Wu, and J. Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *AAAI*, 2021.
- [Chen *et al.*, 2022a] M. Chen, W. Zhang, Z. Yao, X. Chen, M. Ding, F. Huang, and H. Chen. Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In *IJCAI*, 2022.
- [Chen *et al.*, 2022b] M. Chen, W. Zhang, Y. Zhu, H. Zhou, Z. Yuan, C. Xu, and H. Chen. Meta-knowledge transfer for inductive knowledge graph embedding. In *SIGIR*, 2022.
- [Cui *et al.*, 2022] Y. Cui, Y. Wang, Z. Sun, W. Liu, Y. Jiang, K. Han, and W. Hu. Inductive knowledge graph reasoning for multi-batch emerging entities. In *CIKM*, 2022.
- [Cui *et al.*, 2023] Y. Cui, Y. Wang, Z. Sun, W. Liu, Y. Jiang, K. Han, and W. Hu. Lifelong embedding learning and transfer for growing knowledge graphs. In *AAAI*, 2023.
- [Daza *et al.*, 2021] D. Daza, M. Cochez, and P. Groth. Inductive entity representations from text via link prediction. In *WWW*, 2021.
- [Ding *et al.*, 2022] Z. Ding, J. Wu, B. He, Y. Ma, Z. Han, and V. Tresp. Few-shot inductive learning on temporal knowledge graphs using concept-aware information. In *AKBC*, 2022.
- [Galárraga *et al.*, 2013] L. A. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, 2013.
- [Galkin *et al.*, 2022a] M. Galkin, E. G. Denis, J. Wu, and W. L. Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *ICLR*, 2022.
- [Galkin *et al.*, 2022b] M. Galkin, Z. Zhu, H. Ren, and J. Tang. Inductive logical query answering in knowledge graphs. In *NeurIPS*, 2022.
- [Geng *et al.*, 2021] Y. Geng, J. Chen, Z. Chen, J. Z. Pan, Z. Ye, Z. Yuan, Y. Jia, and H. Chen. Ontozsl: Ontology-enhanced zero-shot learning. In *WWW*, 2021.
- [Geng *et al.*, 2022] Y. Geng, J. Chen, W. Zhang, Y. Xu, Z. Chen, J. Z. Pan, Y. Huang, F. Xiong, and H. Chen. Disentangled ontology embedding for zero-shot learning. In *KDD*, 2022.
- [Geng *et al.*, 2023] Y. Geng, J. Chen, W. Zhang, J. Z. Pan, M. Chen, H. Chen, and S. Jiang. Relational message passing for fully inductive knowledge graph completion. In *ICDE*, 2023.
- [Hamaguchi *et al.*, 2017] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *IJCAI*, 2017.
- [He *et al.*, 2020] Y. He, Z. Wang, P. Zhang, Z. Tu, and Z. Ren. VN network: Embedding newly emerging entities with virtual neighbors. In *CIKM*, 2020.
- [Hu *et al.*, 2023] N. Hu, Y. Wu, G. Qi, D. Min, J. Chen, J. Z. Pan, and Z. Ali. An empirical study of pre-trained language models in simple knowledge graph question answering. *CoRR*, 2023.
- [Huang *et al.*, 2022] Q. Huang, H. Ren, and J. Leskovec. Few-shot relational reasoning via connection subgraph pretraining. In *NeurIPS*, 2022.
- [Jiang *et al.*, 2021] Z. Jiang, J. Gao, and X. Lv. Metap: Meta pattern learning for one-shot knowledge graph completion. In *SIGIR*, 2021.
- [Jin *et al.*, 2022] J. Jin, Y. Wang, K. Du, W. Zhang, Z. Zhang, D. Wipf, Y. Yu, and Q. Gan. Inductive relation prediction using analogy subgraph embeddings. In *ICLR*, 2022.
- [Li *et al.*, 2022] X. Li, J. Ma, J. Yu, T. Xu, M. Zhao, H. Liu, M. Yu, and R. Yu. HAPZSL: A hybrid attention prototype network for knowledge graph zero-shot relational learning. *Neurocomputing*, 2022.
- [Liang *et al.*, 2022] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, and F. Sun. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *CoRR*, 2022.
- [Lin *et al.*, 2022] Q. Lin, J. Liu, F. Xu, et al. Incorporating context graph with logical reasoning for inductive relation prediction. In *SIGIR*, 2022.
- [Liu *et al.*, 2021] S. Liu, B. C. Grau, I. Horrocks, and E. V. Kostylev. INDIGO: gnn-based inductive knowledge graph completion using pair-wise encoding. In *NeurIPS*, 2021.



- [Mai *et al.*, 2021] S. Mai, S. Zheng, Y. Yang, and H. Hu. Communicative message passing for inductive relation reasoning. In *AAAI*, 2021.
- [Markowitz *et al.*, 2022] E. Markowitz, K. Balasubramanian, M. Mirtaheri, M. Annavaram, A. Galstyan, and G. Ver Steeg. Statik: Structure and text for inductive knowledge graph completion. In *NAACL-HLT*, 2022.
- [Meilicke *et al.*, 2018] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, and H. Stuckenschmidt. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In *ISWC*, 2018.
- [Meilicke *et al.*, 2019] C. Meilicke, M. W. Chekol, D. Ruffinelli, et al. Anytime bottom-up rule learning for knowledge graph completion. In *IJCAI*, 2019.
- [Niu *et al.*, 2021] G. Niu, Y. Li, C. Tang, R. Geng, J. Dai, Q. Liu, H. Wang, J. Sun, F. Huang, and L. Si. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *SIGIR*, 2021.
- [Pan *et al.*, 2017] J. Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu, editors. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer, 2017.
- [Pan, 2009] J. Z. Pan. Resource Description Framework. In *Handbook of Ontologies*. 2009.
- [Qin *et al.*, 2020] P. Qin, X. Wang, W. Chen, C. Zhang, W. Xu, and W. Y. Wang. Generative adversarial zero-shot relational learning for knowledge graphs. In *AAAI*, 2020.
- [Sadeghian *et al.*, 2019] A. Sadeghian, M. Armandpour, P. Ding, and D. Wang. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *NeurIPS*, 2019.
- [Shah *et al.*, 2019] H. Shah, J. Villmow, A. Ulges, U. Schwannecke, and F. Shafait. An open-world extension to knowledge graph completion models. In *AAAI*, 2019.
- [Sheng *et al.*, 2020] J. Sheng, S. Guo, Z. Chen, et al. Adaptive attentional network for few-shot knowledge graph completion. In *EMNLP*, 2020.
- [Shi and Weninger, 2018] B. Shi and T. Weninger. Open-world knowledge graph completion. In *AAAI*, 2018.
- [Song *et al.*, 2022] R. Song, S. He, S. Zheng, S. Gao, K. Liu, Z. Yu, and J. Zhao. Decoupling mixture-of-graphs: Unseen relational learning for knowledge graph completion by fusing ontology and textual experts. In *COLING*, 2022.
- [Teru *et al.*, 2020] K. K. Teru, E. G. Denis, and W. L. Hamilton. Inductive relation prediction by subgraph reasoning. In *ICML*, 2020.
- [Wang *et al.*, 2017] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *TKDE*, 2017.
- [Wang *et al.*, 2019] P. Wang, J. Han, C. Li, and R. Pan. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *AAAI*, 2019.
- [Wang *et al.*, 2021a] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang. Structure-augmented text representation learning for efficient knowledge graph completion. In *WWW*, 2021.
- [Wang *et al.*, 2021b] H. Wang, H. Ren, and J. Leskovec. Relational message passing for knowledge graph completion. In *KDD*, 2021.
- [Wang *et al.*, 2021c] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and J. Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *TACL*, 2021.
- [Wang *et al.*, 2022a] C. Wang, X. Zhou, S. Pan, L. Dong, Z. Song, and Y. Sha. Exploring relational semantics for inductive knowledge graph completion. In *AAAI*, 2022.
- [Wang *et al.*, 2022b] L. Wang, W. Zhao, Z. Wei, and J. Liu. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *ACL*, 2022.
- [Wang *et al.*, 2022c] Y. Wang, Y. Cui, W. Liu, et al. Facing changes: Continual entity alignment for growing knowledge graphs. In *ISWC*, 2022.
- [Wiharja *et al.*, 2020] K. Wiharja, J. Z. Pan, M. J. Kollingbaum, and Y. Deng. Schema Aware Iterative Knowledge Graph Completion. *Journal of Web Semantics*, 2020.
- [Xie *et al.*, 2016] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun. Representation learning of knowledge graphs with entity descriptions. In *AAAI*, 2016.
- [Xiong *et al.*, 2018] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang. One-shot relational learning for knowledge graphs. In *EMNLP*, 2018.
- [Xu *et al.*, 2021] J. Xu, J. Zhang, X. Ke, Y. Dong, H. Chen, C. Li, and Y. Liu. P-INT: A path-based interaction model for few-shot knowledge graph completion. In *EMNLP (Findings)*, 2021.
- [Xu *et al.*, 2022] X. Xu, P. Zhang, Y. He, et al. Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. In *IJCAI*, 2022.
- [Yan *et al.*, 2022] Z. Yan, T. Ma, L. Gao, Z. Tang, and C. Chen. Cycle representation learning for inductive relation prediction. In *ICML*, 2022.
- [Yang *et al.*, 2017] F. Yang, Z. Yang, and W. W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *NeurIPS*, 2017.
- [Zha *et al.*, 2022] H. Zha, Z. Chen, and X. Yan. Inductive relation prediction by BERT. In *AAAI*, 2022.
- [Zhang and Yao, 2022] Y. Zhang and Q. Yao. Knowledge graph reasoning with relational digraph. In *WWW*, 2022.
- [Zhang *et al.*, 2020] C. Zhang, H. Yao, C. Huang, M. Jiang, Z. Li, and N. V. Chawla. Few-shot knowledge graph completion. In *AAAI*, 2020.
- [Zhang *et al.*, 2021] Y. Zhang, W. Wang, W. Chen, J. Xu, A. Liu, and L. Zhao. Meta-learning based hyper-relation feature modeling for out-of-knowledge-base embedding. In *CIKM*, 2021.
- [Zhang *et al.*, 2023] W. Zhang, Z. Yao, M. Chen, Z. Huang, and H. Chen. Neuralkg-ind: A python library for inductive knowledge graph representation learning. In *SIGIR*, 2023.
- [Zhu *et al.*, 2021] Z. Zhu, Z. Zhang, L. P. Xhonneux, and J. Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *NeurIPS*, 2021.