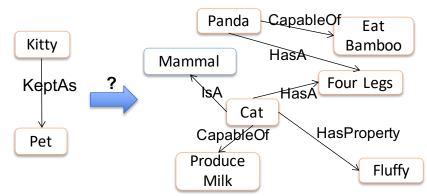
Open-World Taxonomy and Knowledge Graph Co-Learning

Jiaying Lu
jiaying.lu@emory.edu
Emory University
Atlanta, Georgia

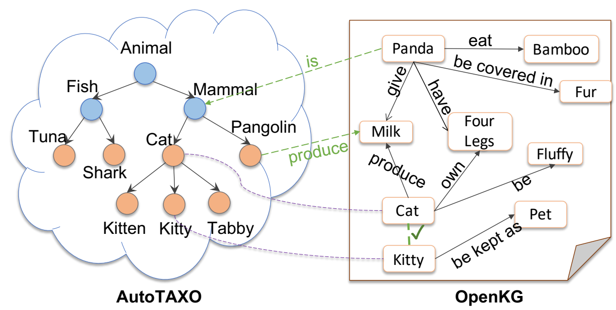
Carl Yang
j.carlyang@emory.edu
Emory University
Atlanta, Georgia

ABSTRACT

Taxonomies and knowledge graphs (KGs), which represent real-world entities' abstract concepts and properties/behaviors/facts, constitute the essential information in knowledge bases (KBs). However, most existing KBs are constructed under the closed-world assumption, which often corresponds to a fixed schema and requires ad-hoc canonicalization to include new knowledge. To empower KBs towards easy accommodation of emerging entities and relations, we propose to create open-world TaxoKG based on existing automatically constructed taxonomies and open KGs, where taxonomies serve to provide a loosely-defined schema and mitigate the reliance on ad-hoc canonicalization. To further improve the completeness of TaxoKG, we collect several new benchmark datasets towards the development of HakeGCN, an innovative hierarchy-aware graph-friendly model for TaxoKG completion. HakeGCN learns to leverage the mutual enhancement between taxonomies and KGs, following the human reasoning process to generalize and conceptualize over taxonomic and non-taxonomic relations. Through extensive experiments, we demonstrate HakeGCN to outperform various state-of-the-art KB completion methods on both taxonomy concept prediction and KG relation prediction tasks based on both standard metrics and human evaluations. The benchmark datasets and the implementation of HakeGCN are available at https://github.com/lujiaying/Open-World-TaxoKG-CoLearning.



(a) Closed-world KB



(b) TaxoKG

Figure 1: Toy examples of existing KBs and TaxoKG.

1 INTRODUCTION

Knowledge bases (KBs) have incorporated large-scale multi-relational data and motivated many knowledge-driven web applications such as online encyclopedia [37] and e-commerce product catalog [9]. The knowledge stored in KBs can be categorized into two types:

(1) The taxonomic knowledge that contains hierarchical *IsA* relations between *entities* and *abstract concepts*, which are typically stored in *taxonomies* (*e.g.*, "*(Cat, IsA, Mammal)*" in Figure 1a);

(2) The non-taxonomic knowledge that contains graph-structured interactions between *entities* and attributes of *entities*, which are often stored in *knowledge graphs* (KGs) (*e.g.*, "*(Cat, HasProperty, Fluffy)*" in Figure 1a).

The taxonomy is a useful tool to organize and index concepts of entities so that users can find the information of interest more easily [29]. On the other hand, the knowledge graph stores human understanding about entities' properties, facts, or behaviors in a structured way, which is essential for knowledge representation and reasoning tasks [8]. Extensive efforts have been made to collect KBs (*e.g.*, WordNet [20], Freebase [1], YAGO [34]) that contain both taxonomies and KGs. However, most existing KBs are in closed domains, and the creation of such KBs highly relies on pre-defined schema [25] and exhaustive entity/relation canonicalization [38]. Although such a creation process ensures accuracy, closed-world KBs are limited in knowledge coverage. As an illustration, when a new knowledge triplet "(*Kitty*, *KeptAs*, *Pet*)" is introduced, although as humans we know kitty has the same meaning as cat, the closed-world KB can not easily incorporate the new knowledge unless the canonicalization tool can directly identify *Kitty* as *Cat*. In conclusion, closed-world KB is most suitable for fixed or slowly evolving knowledge-enhanced applications.

Real-life knowledge management and discovery applications need to evolve with the fast-expanding entities and relations. To accommodate with the new emerging data, we propose to build open-world KBs with both taxonomic relations and non-taxonomic relations, namely TaxoKG, by integrating automatically constructed taxonomies (AutoTAXOs) and open knowledge graphs (OpenKGs). An AutoTAXO is a collection of entity-concept pairs mined from billions of web pages and search logs (*e.g.*, ConceptNet [32], MS Concept Graph [39]), and an OpenKG is a large number of factual triplets collected with open information extraction techniques from unstructured online texts (*e.g.*, ReVerb [10], OPIEC [11]). Figure 1b shows a toy example of TaxoKG. For both AutoTAXO and OpenKG, no fixed schema or ad-hoc canonicalization are required, as the taxonomy provides a loosely-defined underlying schema (*e.g.*, AutoTAXO in Figure 1b naturally defines a schema of animals) and mitigates the reliance on ad-hoc canonicalization (*e.g.*, *Kitten, Kitty, Tabby* are three children entities of *Cat* under open-world KB setting in Figure 1b, while in closed-world setting these three terms need to be canonicalized into one entity).

To empirically understand and show the utility of TaxoKG and its downstream applications, we create and release TaxoKG-Bench, a new benchmark with six datasets covering general, medical, and music domains. To the best of our knowledge, our work is the first one to study the open-world integration of taxonomies and KGs. Although covering an unprecedentedly large amount of entities, concepts and relations, the knowledge in TaxoKG-Bench is not yet fully exploited due to the incompleteness of AutoTAXOs and OpenKGs themselves (*e.g.*, missing edges like "(*Siamese Cat*, *is*, *Cat*)"can be added to AutoTAXO, while "(*cat*, *be covered in*, *fur*)" can be added to OpenKG in Figure 1b). Therefore, it is urgent to develop the corresponding completion method for TaxoKG.

One of the most significant challenges for the open-world TaxoKG completion task is to handle unseen entities, concepts and relations. Previous KB completion methods often rely on the KB embeddings to predict the validity score of missing edges [2, 36, 42, 44]. However, these methods need to add embeddings and re-estimate model parameters when presented with new data. In this work, we address this limitation by utilizing word embeddings of surface tokens for entity, concept and relation embeddings, instead of directly assigning them through the look-up table. Another key insight to complete TaxoKG is to leverage the mutual enhancement between taxonomy and KG. Taxonomies convey rich context on inferring the entities' properties and behaviors (*i.e.*, non-taxonomic relations). For example, as humans, we have the commonsense knowledge of "*mamal can produce milk*". Hence, if we encounter an unseen mammal called "*pangolin*", even if we have no idea what it exactly is, we can confidently infer that "*pangolin can also produce milk*". This reasoning ability is called "generalization" in cognition science [14, 33]. Furthermore, KGs are helpful for deducing entities' abstract concepts (*i.e.*, non-taxonomic relations). If we know that "*mammal can eat, can produce milk, has fur*", and "*pangolin also can eat, can produce milk, and has fur*", it is highly possible that "*pangolin belongs to mammal*". This inference process involves the abilities to *conceptualize*, which is heavily used for information compression and convenient communication [6, 23]. Existing KB completion methods [2, 36, 42, 44], unfortunately, are not designed to leverage such mutual enhancement between taxonomy and KG, thus leaving the jointly learning on TaxoKG an open research problem.

In this work, we propose HakeGCN, a novel hierarchy-aware graph-friendly model which leverages the mutual enhancement between taxonomy and KG. To model the entity hierarchy in taxonomies, HakeGCN employs the polar coordinate KB embedding first proposed in [44], which utilizes the modulus size to reflect depth of the hierarchy and the phase information to represent the entities' surrounding non-taxonomic relations. Moreover, to capture the higher-order relations that are shown to be important for reasoning over KGs [8], HakeGCN utilizes the graph convolutional neural network (GCN)-based knowledge embedding models [26, 36, 43]. A series of technical designs including taxonomy-based neighbor sampling, polar convolution, and GCN-oriented phase bounded decoder is proposed to seamlessly enable the polar embedding system with the GCN architecture. The effectiveness of our proposed HakeGCN is demonstrated through comprehensive experiments on our constructed TaxoKG-Bench. We first examine the model performance on the TaxoKG completion task regarding both AutoTaxo concept prediction and OpenKG relation prediction in terms of the classical metrics and human evaluations, in comparison with several state-of-the-art KB completion models. Then, we conduct extensive ablation studies to evaluate the utility of our technical designs and the impact of neighbor information on mutual enhancement between taxonomy and KG. Finally, we provide case studies for inferred knowledge and analyze the efficiency and scalability of HakeGCN in the Appendix.

2 RELATED WORK

Knowledge Base Completion. KB completion task aims at inferring missing facts based on the known facts. Traditional approaches for KB completion include rule-based inference models [5], path ranking models [18], and probabilistic relational models [17]. One popular approach for KB completion is to embed entities and relations into vector spaces, and to define a score function such that valid triples are assigned a higher score than the invalid ones. These KB embedding methods can be categorized into translation-based models [2, 35, 44], tensor factorization-based models [22, 42], and neural network-based models [7, 26, 36]. More recently, HAKE [44], inspired by RotatE [35], utilizes the modulus and phase information in modeling entities and relations. On the other hand, RGCN [26] and CompGCN [36] incorporate the powerful graph neural networks as the encoder to propagate the relation-specific information among interlinked entities and utilize translational scoring function as decoder to infer the validity of edges.

Open-World Knowledge Bases. Existing KB completion models implicitly follow the closed-world assumption [24] in which all entities and relations have been observed and only missing links of known relations between existing entities can be discovered. Unfortunately, closed-world KB completion models fail to adapt to new emerging entities and relations in many real-life applications [31]. It is of interest to infer knowledge about entities and relations not present in the existing KB, which is known as open-world KB completion [3, 12, 27]. CaRe [12] propose a canonicalization-infused representation model to enrich OpenKB embeddings with the output of a canonicalization model, while OWE [27] predict facts for unseen entities based on their textual description. Inspired by Complex-LSTM [3], our proposed HakeGCN model utilizes the word embeddings to construct entity and relation embeddings, hence no extra resources are needed to handle unseen data.

Co-Learning of Taxonomy and Knowledge Graph. Previous works about taxonomy mainly focus on automatic taxonomy construction [30] and taxonomy-guided downstream tasks [29, 40]. On the other side, extensive efforts have been put on KG construction [1, 34], KG completion [2, 36], and KG-enhanced applications [8, 15]. Although there exist attempts to collect closed-world KBs that contain both taxonomies and KGs [1, 20, 32, 34], taxonomies and KGs in the open-world setting (AutoTAXO and OpenKG) have rarely been studied together [13, 41]. JOIE [13] proposes a universal representation of entities and concepts for a two-view KB, which contains the ontology-view KG and the instance-view KG. GeoAlign [41] utilizes the manifold-aligned hyperbolic embedding for taxonomy and Euclidean embedding for KG to tackle the KB representation learning problem. Both JOIE and GeoAlign are designed for closed-world KBs, thus not directly applicable to our open-world KB setting.

3 PROBLEM DEFINITION

Given an open-world KB — TaxoKG $\mathcal{B}$ containing the taxonomy $\mathcal{T}$ and the knowledge graph $\mathcal{G}$, the TaxoKG co-learning (or completion) task aims to jointly infer the missing edges in $\mathcal{T}$ and $\mathcal{G}$. The AutoTAXO $\mathcal{T} = (\mathcal{V}_e, \mathcal{V}_c, \mathcal{E}_\mathcal{T})$ is a collection of entity-concept pairs, where $\mathcal{V}_e$ and $\mathcal{V}_c$ are entity and concept sets, and $\mathcal{E}_\mathcal{T} = \{(e, c)\} \subseteq \mathcal{V}_e \times \mathcal{V}_c$ is the set of the taxonomic edges. There are some terms in the intersection between $\mathcal{V}_e$ and $\mathcal{V}_c$, since a term ("*mammal*") can serve as a hyponym (*e.g.*, *(mammal, animal)*) and a hypernym (*e.g.*, *(cat, mammal)*) at the same time. Moreover, taxonomic edges indicates the *IsA* relations between entities and concepts (either *IsA_InstanceOf* or *IsA_SubClassOf*). The OpenKG $\mathcal{G} = (\mathcal{V}_e, \mathcal{R}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ is a collection of subject-relation-object triplets, where $\mathcal{V}_e$ is the entity set identical to $\mathcal{T}$'s entity set, $\mathcal{R}_\mathcal{G}$ is the relation set that contains many types of relations (excluding taxonomic relations), and $\mathcal{E}_\mathcal{G} = \{(s, r, o)\} \subseteq \mathcal{V}_e \times \mathcal{R}_\mathcal{G} \times \mathcal{V}_e$ is the edge set connecting entities with associated relations. Therefore, the TaxoKG containing $\mathcal{T}$ and $\mathcal{G}$ can be denoted as $\mathcal{B} = (\mathcal{V}_e, \mathcal{V}_c, \mathcal{R}_\mathcal{G}, \mathcal{E}_\mathcal{T}, \mathcal{E}_\mathcal{G})$.

The TaxoKG completion task is a variant of the general open-world KB completion task, which can defined as follows:

Definition 3.1 (Open-world KB Completion)**.** Given the incomplete KB $\mathcal{B} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ where $\mathcal{V}$, $\mathcal{R}$ and $\mathcal{E}$ are entity set, relation set and triplet set, open-world KB completion aims at inferring the missing triples $\{(s, r, o) | (s, r, o) \notin \mathcal{E}, s \in \mathcal{V}^s, r \in \mathcal{R}^s, o \in \mathcal{V}^s\}$, where $\mathcal{V}^s$ and $\mathcal{R}^s$ are entity superset and relation superset.

Hence, the TaxoKG can be generalized to a big directed and multi-relational graph $\mathcal{B} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_e \cup \mathcal{V}_c$, $\mathcal{R} = \mathcal{R}_\mathcal{G} \cup \mathcal{R}_{\text{IsA}}$ ($\mathcal{R}_{\text{IsA}}$ denotes the specific taxonomic relation in $\mathcal{T}$), and $\mathcal{E} = \mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{T}$. More specifically, there are two sub-tasks for TaxoKG completion: (1) the AutoTAXO concept prediction task and (2) the OpenKG relation prediction task. The AutoTAXO concept prediction task is to assign a set of concepts $C_e = \{c_1, c_2, \ldots, c_m\}$ for each entity $e \in \mathcal{V}_e$, while OpenKG relation prediction aims to predict missing facts in the form of $q_s = (?, r_k, o_j)$ or $q_o = (s_i, r_k, ?)$. It is worth noting that $e, s, o \in \mathcal{V}_e^s$, $c \in \mathcal{V}_c^s$, and $r \in {\mathcal{R}_\mathcal{G}}^s$, which indicate that we need to handle unseen entities, concepts or relations.

4 TAXOKG-BENCH: A NEW BENCHMARK WITH SIX DATASETS FOR TAXOKG COMPLETION

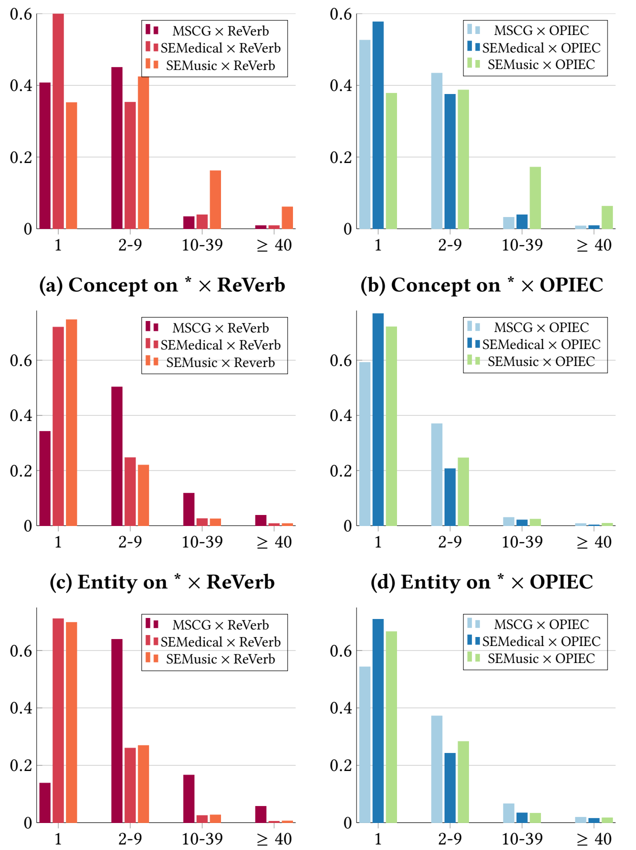
4.1 Source Data

The goal of building TaxoKG-Bench[1] is to provide a benchmark to evaluate models' performance on TaxoKG completion task, which involves the ability to predict new-emerging concepts and novel facts for unseen entities. The source data includes:

- Three AutoTAXOs: MS Concept Graph [39], SemEval-2018 Task 9 2A:Medical and 2B:Music [4];
- Two OpenKGs: ReVerb [10] and OPIEC [11].

MS Concept Graph (MSCG) is a large-scale AutoTAXO that contains millions of entity-concept pairs from billions of web pages, while SemEval-2108 Task 9 Medical (SEMedical) and Music(SEMusic) are two small-scale AutoTAXOs containing thousands of entity-concept pairs constructed from medical and music domain corpora. On the other side, both ReVerb and OPIEC are OpenKGs that consist of a massive amount of subject-relation-object triplets extracted from English web pages and Wikipedia. Since AutoTAXOs and OpenKGS exhaustively extract ontology-relations and instance-relations from text, the knowledge triplets stored in TaxoKG are numerous and not constrained by the finite schema. Moreover, all entity, concept, and relation mentions are not canonicalized, thus introducing more challenges to the TaxoKG completion task.

4.2 Creation Process



(a) Concept on * × ReVerb (b) Concept on * × OPIEC
(c) Entity on * × ReVerb (d) Entity on * × OPIEC
(e) Relation on * × ReVerb (f) Relation on * × OPIEC

Figure 2: Concept, entity and relation histograms on six aligned TaxoKGs.

We first align AutoTAXOs and OpenKGs by matching entities in entity-concept pairs with subjects or objects in subject-relation-object triplets. It is possible to use off-the-shelf entity canonicalization tools [38] to match entities. However, their reliabilities are far away from satisfactory. Consequently, we just use naive string matching to align AutoTAXOs and OpenKGs. As can be seen from Figure 2, the distribution of concepts, entities, and relations basically follow Zipf's law. In other words, there are many long-tailed concepts, entities, and relations that only show one or two times. Since our target is not to evaluate the TaxoKG completion task under a zero-or-few-shot setting, we decide to discard these long-tailed concepts, entities, and relations. To build our benchmark

[1] We release TaxoKG-Bench: https://figshare.com/articles/dataset/Taxo-KG-Bench/16415727

efficient and easy access to researchers, we conduct further post-processing (Appendix A) on the aligned six datasets to obtain the final release version.

4.3 Benchmark Overview

The Statistics. After creation process mentioned above, we obtain the final version of six TaxoKGs, as can be seen in Table 5 under Appendix A. For AutoTAXOs side, we split the entity-concept pairs by randomly assigning 55%, 5%, 35% entities into training, validation, testing set. For OpenKG side, we split subject-relation-object triplets by randomly assigning 80%, 5%, 15% triplets into training, validation, testing set. In other words, each split set is the union of assigned ontology-relation set and instance-relation set.

Comparison to Other KBs. We aim to construct TaxoKG-Bench as a large-scale, diverse, challenging benchmark for open-world KB completion. The coverage of TaxoKG-Bench is broad since the data sources come from various genres: general, medical, and music domain. The sizes also range from thousands to hundred-thousands knowledge triplets (including both entity-concept pairs and subject-predicate-object triplets). Moreover, the scale of relations is much larger than previous closed-world KBs. For instance, FB15k-237 [1] contains 237 unique relations, while WN188RR [20] contains only 11 relations. Consequently, such a large relation space requires stronger KB completion models. The proportions of ontology knowledge in TaxoKG-Bench vary from 1.4% to 13.5%, and thus enriches the benchmark's diversity.

Unseen Concepts, Entities, Relation in TaxoKG-Bench Our TaxoKG-Bench is different from existing KBs due to the open-world setting and the integration of taxonomy and KG. As a result, significant portions of entities, concepts, and relations in the test set are not observed in the training set, as opposed to the assumption of closed-world KB that all entities and relations are fixed —only missing edges between existing entities are to be discovered. Table 6 under Appendix A shows the percentages of unseen entities, concepts and relation in the six TaxoKGs. In the most challenging one MSCG × OPIEC, nearly half of the entities, relations, and one-third of concepts are hidden during training, which poses a serious challenge for models targeted at the *Taxo-KG* completion task.

5 HAKEGCN: A NOVEL METHOD FOR EFFECTIVE TAXOKG COMPLETION

5.1 Overall Model Design

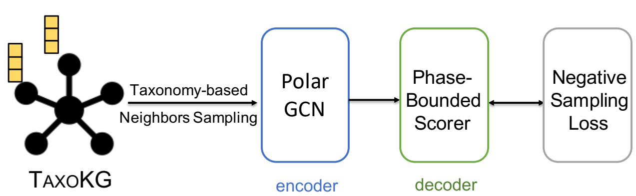


Figure 3: HakeGCN model architecture.

To tackle the TaxoKG completion task, our key insight is to leverage the mutual enhancement between taxonomy and KG. Therefore, we propose a novel model with the learn-to-conceptualize and learn-to-generalize abilities via combining advantages from **H**ierarchy-**A**ware **K**nowledge base **E**mbedding [44] and **G**raph **C**onvolutional neural **N**etworks [36], namely HakeGCN. HakeGCN can be regarded as an encoder-decoder model consisting of (1) an encoder: a multi-relational GCN producing latent representation of vertices (entities and concepts) and edges (relations) in the polar coordinate system; (2) a decoder: a hierarchy-aware translational distance model exploiting these representations to predict labeled edges (entity-concept edges, entity-entity edges). The overall model architecture is shown in Figure 3.

5.2 Handling Unseen Entities, Concepts and Relations

As described in §4.3, there are a significant number of unseen entities, concepts, and relations that need to be handled in the TaxoKG completion task. Unfortunately, most existing models for KB completion [2, 26, 44] are developed under the closed-world setting, therefore their solution to embed phrases is to assign a look-up embedding table and update the embeddings during the training phase. As a consequence, they fail to obtain embeddings for new emerging phrases in the open-world setting. In HakeGCN, we opt to create entity, concept, and relation representation from the tokens of the surface mentions [3]. The entity and concept representations are then fed into the GCN encoder as initial embeddings of vertices, and relation representations as initial embeddings of edges. Therefore, for any vertex or edge h that is in the form of a sequence of tokens $\{t_1, t_2, \ldots, t_L\}$, the representation is calculated by

$$\boldsymbol{h} = f(h) = f_{phr}(f_{tok}(t_1), f_{tok}(t_2), \ldots, f_{tok}(t_L)), \tag{1}$$

where the lowercase letter h denotes vertex or edge phrase, the boldface lowercase letter $\boldsymbol{h}$ denotes the phrase embedding of vertex or edge, $f_{tok} : \mathbb{V}^{Tok} \to \mathbb{R}^d$ denotes the token embedding look-up mapping function, and $f_{phr} : \mathbb{R}^{L\times d} \to \mathbb{R}^{d'}$ denotes the phrase composition function. The choice of composition functions is flexible, which includes *average*, *sum*, *max*, *RNN* and even *Transformer*. In HakeGCN, we choose *average* for the sake of simplicity. The token embedding look-up table is shared among vertices and edges.

After taking the average of token embeddings, we apply different single-layer perceptrons on $\boldsymbol{h_v}, \boldsymbol{h_r}$ to obtain the vertex and edge embeddings separately:

$$\boldsymbol{h}_v^0 = \mathrm{PReLU}(\boldsymbol{W_v h_v} + \boldsymbol{b_v}) \quad \text{and} \quad \boldsymbol{h}_r^0 = \mathrm{PReLU}(\boldsymbol{W_r h_r} + \boldsymbol{b_r}). \tag{2}$$

Here, we use v to represent any entity $e \in \mathcal{V}$ and concept $c \in \mathcal{V}$ that can be viewed as the vertex of the overall knowledge base $\mathcal{B} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$. Similarly, we use r to represent the IsA relation $\mathcal{R}_{\text{IsA}} \in \mathcal{R}$ of AutoTaxo and any relation $r \in \mathcal{R}$ of OpenKG that can be viewed as the edge of $\mathcal{B}$. For the non-linear activation, we opt to PReLU [16]. The superscript 0 denotes that we use them as the input of the GCN encoder.

5.3 GCN Encoder with Polar Convolution and Taxonomy-based Neighbor Sampling

Our novel encoder is a generalization of inductive GCN encoders in polar coordinates that benefits from the expressiveness of both graph neural networks and hierarchy-aware polar embeddings. First, since the input vertex and edge embeddings are Cartesian coordinate embeddings, we derive the relational neighbor aggregation and embedding updating in the Cartesian system. Next, we derive a mapping from Cartesian coordinates to polar coordinates. We finally use the mapped polar embeddings of vertices and edges as the input of the decoder.

Updating Embeddings in Cartesian Coordinate. Since the input features or initial embeddings are often represented in the Cartesian coordinate system, we first conduct the widely-studied relation-GCNs defined in the Cartesian coordinate system on these embeddings. The choice of the relational-GCN encoder is flexible, as long as it takes both vertex and edge representations into account. We propose our own node updating rules:

$$\boldsymbol{m}_{\boldsymbol{v}}^{\boldsymbol{k+1}} = \textsc{Agg}(\{\boldsymbol{W}_{\boldsymbol{dir(r)}}^{\boldsymbol{k}}\ \phi(\boldsymbol{h}_{\boldsymbol{u}}^{\boldsymbol{k}}, \boldsymbol{h}_{\boldsymbol{r}}^{\boldsymbol{k}}), \forall (u,r) \in \mathcal{N}(v)\}), \tag{3}$$

$$\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{k+1}} = \mathrm{PReLU}(\boldsymbol{W}_{\boldsymbol{v}}^{\boldsymbol{k}}[\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{k}} \parallel \boldsymbol{m}_{\boldsymbol{v}}^{\boldsymbol{k+1}}] + \boldsymbol{b}_{\boldsymbol{v}}^{\boldsymbol{k}}). \tag{4}$$

In Eq. (3), the message on vertex v is collected from the neighbors $N(v)$. The composition function $\phi(\boldsymbol{h_u}, \boldsymbol{h_r})$ can be either $\boldsymbol{h_u} - \boldsymbol{h_r}$, $\boldsymbol{h_u} * \boldsymbol{h_r}$ or $\boldsymbol{h_u} \star \boldsymbol{h_r}$, where operator $\star$ denotes the circular correlation. The aggregation function $\textsc{Agg}(\cdot)$ can be choose freely from *average*, *sum*, *max* or other functions. Specifically, the relation-specific learnable parameter in Eq. (3) is

$$\boldsymbol{W}_{\boldsymbol{dir(r)}} = \begin{cases} \boldsymbol{W_O}, & (u,r,v) \in \mathcal{E}, \\ \boldsymbol{W_I}, & (u,r,v) \in \mathcal{E}_{inv}, \end{cases} \tag{5}$$

where $\mathcal{E}_{inv}$ denotes invert edges introduced to $\mathcal{B}$ for better vertex and edge representation. In Eq. (4), the message representation $\boldsymbol{m}_{\boldsymbol{v}}^{\boldsymbol{k+1}}$ is first concatenated with the node representation $\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{k}}$, and then feed into a multi-layer perceptron. The relation-specific learnable parameter is similar to CompGCN [36]. Our novel design is that we do not introduce self-loops during message aggregation, but concatenate the self node embeddings with aggregated neighborhood embedding during node representation updating. Moreover, the edge updating rule is:

$$\boldsymbol{h}_{\boldsymbol{r}}^{\boldsymbol{k+1}} = \mathrm{PReLU}(\boldsymbol{W}_{\boldsymbol{r}}^{\boldsymbol{k}} \boldsymbol{h}_{\boldsymbol{r}}^{\boldsymbol{k}} + \boldsymbol{b}_{\boldsymbol{r}}^{\boldsymbol{k}}). \tag{6}$$

Eq. (6) is only used to update edges in the training graph during the encoding phase, the representation of relation r for predicting knowledge triplet (s,r,o) is calculated through another similar transformation in the decoder.

Mapping from Cartesian to Polar Representations. The neighborhood aggregation and updating operations in the GCN encoder of HakeGCN are defined in the Cartesian coordinate system, while the hierarchy-aware decoder works in the polar coordinate system. To bridge the gap between these two coordinate systems, we map entity and relation embeddings from Cartesian coordinate to polar coordinate, using the following equations:

$$\rho = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \mathrm{atan2}(y,x), \tag{7}$$

where $x, y \in \mathbb{R}$, $\rho \in \mathbb{R}_+$, and $\theta \in [-\pi, +\pi]$. The atan2 function is a common variation of the arctangent function. During the polar convolution process above, vertex and edge embeddings in Cartesian coordinate can be denoted as $\boldsymbol{h} = [\boldsymbol{x} \parallel \boldsymbol{y}]$. Assuming $\boldsymbol{h}$'s dimension is $2d$, then $\boldsymbol{h}$ stores d pairs of Cartesian coordinates. Therefore, using Eq. (7), $\boldsymbol{h}$ can be mapping into the an embedding containing d pairs of polar coordinates, denoted as $\boldsymbol{h} = [\boldsymbol{\rho} \parallel \boldsymbol{\theta}]$.

Taxonomy-based Neighborhood Sampling. Although the neighborhood information is generally useful, many existing GCN-based models keep all neighbors during training which introduces noisy and even hazardous information [36, 43]. For instance, presented "platypus is a mammal but lays eggs", GCN-based models may induct that laying eggs is a positive factor to judge an animal belongs to the mammal category. To relieve the noisy neighborhood information, RGCN [26] proposes to apply edge dropout on its encoder. However, the edge dropout process randomly masks out neighbors connected by edges, which may discard useful information. Therefore, we propose a taxonomy-based neighbor sampling strategy. Instead of uniformly sampling over all edges, we propose to assign a higher chance to keep edges between the entity of interest and the neighbors connected by both entity-entity edges and entity-concept edges. The intuition is to allow the GCN to see more neighbors on the taxonomy, which contains less noise. The value of higher chance is chosen through hyper-parameter tuning (Appendix C).

5.4 GCN-Oriented Phase Bounded Decoder

After getting the entity and relation representations from the GCN encoder of HakeGCN, the decoder scores "(subject, relation, object)" triplets through a function $f(s,r,o): \mathbb{R}^d \times \mathbb{R}^{d'} \times \mathbb{R}^d \to \mathbb{R}$, where d and d' denotes the embedding sizes for entity/concept and relation, respectively. Possible decoders include translational models, tensor factorization models, neural network models. In practise, we adapt the HAKE score function with necessary modifications:

$$f(s,r,o) = -d(s,r,o) = -\lambda_m d_m(s,r,o) - \lambda_p d_p(s,r,o). \tag{8}$$

Here, (s,r,o) denotes both entity-concept pairs (associated relation is "IsA" relation) and entity-relation-entity triplets in TaxoKG, and $d(s,r,o)$ denotes the distance function between subject entity s and object entity o in condition of relation r. In particular, $\lambda_m, \lambda_p \in \mathbb{R}$ are two learnable parameters to balance the modulus distance with the phase distance. Moreover, these two distance functions are given by the following equations in HAKE:

$$d_m(s,r,o) = \left\| \boldsymbol{h}_{\boldsymbol{s,m}} \circ \boldsymbol{h}_{\boldsymbol{r,m}} - \boldsymbol{h}_{\boldsymbol{o,m}} \right\|_2, \tag{9}$$

$$d_p(s,r,o) = \left\| \sin((\boldsymbol{h}_{\boldsymbol{s,p}} + \boldsymbol{h}_{\boldsymbol{r,p}} - \boldsymbol{h}_{\boldsymbol{o,p}})) \right\|_1, \tag{10}$$

where $\boldsymbol{h_s}, \boldsymbol{h_o}$ denote the subject, object embeddings obtained from the GCN encoder production $\boldsymbol{h_u}$ in Eq. (4), and $\boldsymbol{h_r}$ denotes the relation embedding obtained from a separate transformation in decoder using a similar process as in Eq. (6). For the polar coordinate, $\boldsymbol{h}_{*,\boldsymbol{m}}, \boldsymbol{h}_{*,\boldsymbol{p}}$ denote the embeddings in the modulus and phase part. In Eq. (9), the operator $\circ$ denotes the Hadamard product between two vectors. Let $\Delta\boldsymbol{\theta} = \boldsymbol{h}_{s,p} + \boldsymbol{h}_{r,p} - \boldsymbol{h}_{o,p}$. In the original phase distance function of HAKE, there is a denominator 2 for $\Delta\boldsymbol{\theta}$, which leads Eq. (10) to $\left\|\sin(\frac{\Delta\boldsymbol{\theta}}{2})\right\|$. This is due to $\boldsymbol{h}_{*,p} \in [0, 2\pi)^d$, and thus $\Delta\boldsymbol{\theta} \in [0, 4\pi)^d$. In our own version of the phase part distance function, we remove the denominator. Therefore, the $\boldsymbol{h}_{*,p}$ produced by atan2 is bounded in $[-\frac{\pi}{2}, +\frac{\pi}{2}]$. This modification is essential because the phase boundary amplifies triplets' phase distances, thus making it easier for decoder to distinguish entities at the same level of the taxonomy.

5.5 Loss function

We adopt the widely used negative sampling loss functions [2, 22, 42, 44] with self-adversarial training [35]:

$$L = -\log\sigma(\gamma - d(s,r,o)) - \sum_{i=1}^{n} p(s'_i, r, o'_i)\log\sigma(d(s'_i, r, o'_i) - \gamma), \tag{11}$$

where σ is the sigmoid function, γ is a fixed margin that can be chosen by hyper-parameter tuning, and (s'_i, r, o'_i) represents the ith sampled negative triplet of (s,r,o). The term $p(s'_i, r, o'_i)$ is the sampling probability of the particular negative triplet, which can be calculated by:

$$p(s'_i, r, o'_i) = \frac{\exp(\alpha f_{samp}(s'_i, r, o'_i))}{\sum_j \exp(\alpha f_{samp}(s'_j, r, o'_j))}, \tag{12}$$

where α is another hyper-parameter that represents the temperature of negative sampling.

6 EXPERIMENTS

In this section, we evaluate our proposed HakeGCN model focusing on the following research questions:

- *RQ1*: How does HakeGCN perform in comparison to other state-of-the-art KB completion methods?
- *RQ2*: Is HakeGCN inferred knowledge reasonable?
- *RQ3*: What are the effects of HakeGCN technical designs?

Moreover, we provide case studies to illustrate how taxonomy and KG mutually enhance each other for the TaxoKG completion task (Appendix D). Furthermore, we evaluate the inference efficiency of HakeGCN and other models (Appendix E).

6.1 Experiment Setting

Datasets & Evaluation Protocols. We evaluate TaxoKG completion task on the proposed TaxoKG-Bench. In TaxoKG-Bench, six datasets covers the general, medial and music domains. There are two sub-tasks in the TaxoKG completion task. For AutoTAXO concept prediction, we aim to assign correct concepts to each entity of interest. This is a multi-label prediction task, and we make the models predict a ranked list of all candidate concepts. Following previous concept assignment task [4], we choose *Mean Average Precision* (MAP) and *Precision at N* (P@N) as evaluation metrics. MAP is calculated based on the top-15 predicted concepts. For OpenKG relation prediction, the goal is to predict either the subject s given query $(?, r, o)$ or the object o given $(s, r, ?)$. We follow previous KB completion studies [2, 12] to rank candidate entities under the "filtered" evaluation protocol, and we choose *Mean Reciprocal Rank* (MRR) and *Hits at N* (H@N) as the evaluation metrics.

Compared Methods. We adopt the following representative state-of-the-art methods as baselines for performance comparison:

- Translation-based models: TransE [2], HAKE [44].
- Tensor factorization models: DistMult [42], HolE [22].
- GCN-based models: R-GCN [26], CompGCN [36].

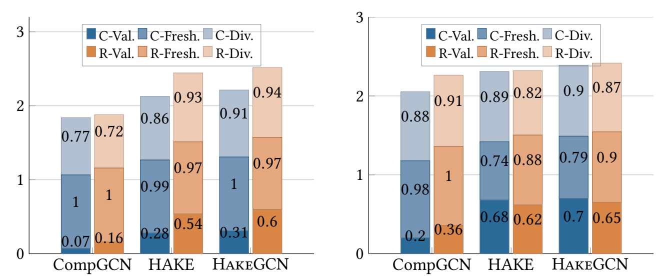
We integrate the same technique introduced in §5.2 to mitigate unseen entities, concepts, and relations in the open-world setting. Moreover, we propose a non-parametric LtCaG model as another baseline method. LtCaG is also inspired by the insight of taxonomy and KG mutual enhancement, and it does not require any training process (See details at Appendix C).

6.2 Performance Comparison (*RQ1*)

We compare the proposed HakeGCN with other baselines through common evaluation metrics used in KB completions. Tables 1, 2, 3 show the performance of HakeGCN and several compared models towards the TaxoKG completion task in general, medical, music domains. Our naive LtCaG model simply relies on cases seen in the training set and intuitive inferring formulas without learnable parameters, but surprisingly achieves competitive performance to all complicated models except for HAKE in AutoTAXO concept prediction metrics (MAP, P@10,30,50) on all six datasets. However, LtCaG performs badly in terms of OpenKG relation prediction metrics (MRR, H@10,30,50). Our HakeGCN significantly outperforms existing state-of-the-art models on all datasets on both tasks, which demonstrates the substantial advantages of integrating taxonomy and KG to mutually complete each other. Other than HakeGCN, HAKE is the second-best model, which only surpasses HakeGCN in P@3,10 metrics in two medical domain datasets.

All baselines except for HAKE fail to complete TaxoKG in all six datasets. There is an obvious pattern when entity and relation numbers grow from hundreds in SEMedical × OPIEC to ten-thousands in MSCG × ReVerb, their performance drops significantly due to the ignorance on handling unseen entities and relations. HAKE performs better than other previous models in all six datasets, as it utilizes the polar coordinate embedding to model the hierarchy-aware property of TaxoKG. The proposed HakeGCN further improves the performance by adding the reasoning ability over higher-order evidence upon the polar coordinate embedding.

6.3 Human Evaluation (*RQ2*)



(a) MSCG × ReVerb **(b) SEMusic × ReVerb**

Figure 4: Qualitative evaluation for different models.

We conduct qualitative evaluations including human-annotations on knowledge triplets generated by two strongest baseline models CompGCN, HAKE, and our HakeGCN, using the following metrics:

- Validity (Val.): We ask annotators to examine whether generated knowledge triplets are valid to humans.
- Freshness (Fresh.): We use the percentage of generated knowledge triplets that are novel (not present in TaxoKG).

Table 1: Experiment results on TaxoKG completion in the general domain.

	MSCG × ReVerb				MSCG × OPIEC			
	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50
TransE	.007	.001, .003, .002	7e-4	8e-4, .002, .004	.006	.004, .002, .001	.002	.001, .004, .008
HAKE	.034	.013, .013, .010	.029	**.065**, **.120**, **.153**	.031	.014, .011, .010	.539	**.787**, **.821**, **.837**
DistMult	.004	.004, .001, 5e-4	.001	3e-4, .004, .006	.001	9e-4, 3e-4, 3e-4	.080	.131, .159, .176
HolE	.007	.003, .003, .002	7e-4	7e-4, .002, .004	.006	.004, .002, .001	.002	.001, .004, .008
R-GCN	.003	5e-4, .001, 8e-4	.001	8e-4, .003, .007	.044	.044, .017, .006	.017	.031, .121, .179
CompGCN	.014	.008, .005, .004	4e-4	2e-4, 6e-4, 8e-4	.004	.003, .002, .001	.011	.025, .051, .067
LtCaG	.005	.003, .002, .002	.001	.002, .003, .004	.003	.002, .001, .001	.002	.002, .006, .009
HakeGCN	**.069**	**.033**, **.028**, **.017**	**.031**	.058, .113, .150	**.070**	**.052**, **.027**, **.014**	**.675**	.756, .805, .832

Table 2: Experiment results on TaxoKG completion in the medical domain.

	SEMedical × ReVerb				SEMedical × OPIEC			
	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50
TransE	.036	.104, .083, .050	.002	.002, .009, .012	.025	.045, .061, .030	.005	.007, .019, .030
HAKE	.203	.307, **.286**, **.216**	.170	.343, .430, .459	.262	.371, .309, **.256**	.352	.450, .509, .544
DistMult	.065	.188, .069, .033	.023	.070, .135, .187	.022	.159, .068, .032	.032	.061, .158, .218
HolE	.029	.063, .063, .044	.002	.002, .005, .009	.024	.091, .030, .027	.006	.007, .018, .032
R-GCN	.024	.018, .041, .052	.001	.001, .003, .004	.036	.159, .062, .037	.004	.003, .016, .026
CompGCN	.119	.191, .184, .150	.003	.005, .012, .017	.041	.060, .044, .032	.009	.013, .023, .034
LtCaG	.186	.245, .247, .172	.004	.005, .006, .008	.126	.166, .157, .122	.013	.021, .041, .051
HakeGCN	**.233**	**.331**, .278, .204	**.275**	**.424**, **.545**, **.603**	**.271**	**.377**, **.366**, .251	**.412**	**.508**, **.600**, **.652**

Table 3: Experiment results on TaxoKG completion in the music domain.

	SEMusic × ReVerb				SEMusic × OPIEC			
	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50	C-MAP	C-P@1, 3, 10	R-MRR	R-H@10, 30, 50
TransE	.012	.053, .035, .028	.002	.002, .006, .009	.041	.123, .082, .064	.002	.003, .008, .013
HAKE	.201	.275, .270, .210	.131	.258, .344, .382	.284	.379, .363, .294	.321	.497, .612, .669
DistMult	.035	.118, .092, .066	.019	.039, .123, .188	.047	.086, .078, .081	.017	.044, .092, .124
HolE	.038	.118, .092, .066	.002	.002, .004, .007	.028	.062, .066, .043	.003	.003, .008, .015
R-GCN	.005	.011, .010, .013	8e-4	7e-4, .002, .003	.014	.021, .039, .034	.002	.001, .005, .008
CompGCN	.063	.092, .111, .095	.009	.019, .034, .042	.082	.199, .161, .112	.005	.012, .023, .036
LtCaG	182	.286, .251, .172	.003	.004, .006, .009	.287	**.426**, .378, .251	.025	.040, .055, .063
HakeGCN	**.238**	**.301**, **.307**, **.221**	**.178**	**.286**, **.412**, **.481**	**.328**	**.426**, **.417**, **.310**	**.421**	**.572**, **.694**, **.746**

- Diversity (Div.): We use Pielou's evenness index[2] which is a commonly used diversity index to represent how close in numbers each species in an environment is.

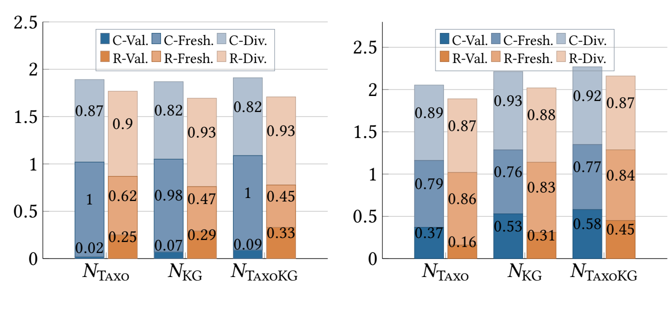
We collect results and compute the three metrics on the AutoTAXO concept prediction task by the top-5 predicted concepts, given 100 entities from MSCG × ReVerb and SEMusic × ReVerb. Similarly, we collect results on OpenKG link computed from the top-5 predicted subject or object entities, given 100 triplet queries. As shown in Figure 4, the blue stacked bar contains the freshness (C-Fresh.), validity (C-Val.), and diversity (C-Div.) of concepts assigned to entities to predict, and the orange stacked bar contains the freshness (R-Fresh.), validity (R-Val.) and diversity (R-Div.) of generated open knowledge triplets. Therefore, HakeGCN produces the highest quality knowledge triplets. In particular, HakeGCN outperforms the two baseline models in both taxonomy and KG validity, with competitive freshness and diversity.

6.4 Ablation Study (*RQ3*)

Do our technical designs contribute to performance boost? To better understand our proposed techniques, we closely study the key components of HakeGCN. The three components are: taxonomy-based neighbors sampling (§5.3), polar GCN (§5.3), and GCN-oriented phase bounded decoder (§5.4). Table 4 presents the results on two medical TaxoKG's with the major metrics for both the AutoTAXO concept prediction and the OpenKG relation prediction tasks. From the table, we can see that all three components improve the performance of HakeGCN, which illustrates the effectiveness of the proposed techniques.

Table 4: Ablation study results.

Model	SEMedical × ReVerb		SEMdical × OPIEC	
	C-MAP	R-MRR	C-MAP	R-MRR
HakeGCN	**.233**	**.275**	**.271**	**.412**
w/o. taxo_graph_sampling	.154	.268	.151	.376
w/o. polar_conv	.155	.254	.196	.331
w/o. phase_bounded_scorer	.152	.239	.216	.311



(a) MSCG × OPIEC **(b) SEMedical × ReVerb**

Figure 5: Qualitative evaluation for neighborhood impact.

[2]Pielou's eveness index: https://en.wikipedia.org/wiki/Species_evenness

Can taxonomy and KG mutually enhance each other? We further analyze the impact of neighbor information from AutoTAXOs and OpenKGs. In Figure 5, we plot the human evaluation results of HakeGCN when using neighbors on AutoTAXOs alone (N_{Taxo}), OpenKGs alone (N_{KG}), and both AutoTAXOs and OpenKGs (N_{TaxoKG}). For the GCN encoder, N_{Taxo} is implemented by removing all taxonomic relation edges in the input graph, and N_{KG} by removing all non-taxonomic relation edges. The metrics and notations are the same as Figure 4. As can be seen from Figure 5, using only one type of neighbors does not significantly impact the freshness and diversity. In contrast, using both types of neighbors from taxonomy and KG can produce more valid knowledge triplets (*e.g.* improving from 0.02/0.07 to 0.09 in MSCG × OPIEC and from 0.37/0.53 to 0.58 in SEMedical × ReVerb). Such results clearly demonstrate the substantial mutual enhancement between the taxonomies and KGs towards the completion of TaxoKG.

7 CONCLUSIONS

To address the rigidity of closed-world KBs, we propose to construct TaxoKG by integrating automatically constructed taxonomies and KGs in the open-world setting. A benchmark TaxoKG-Bench with six datasets is created and released for developing and evaluating models for the novel TaxoKG completion task. Experiments on the benchmark show that our novel hierarchy-aware and graph-friendly KB completion model, HakeGCN, can effectively complete TaxoKG to further improve its knowledge coverage with good validity, so as to better support various knowledge-enhanced applications with the need of rapidly evolving knowledge. In the future, it would also be interesting to further improve HakeGCN by explicitly leveraging the mutual enhancement between taxonomic and non-taxonomic knowledge.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*.

[3] Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can We Predict New Facts with Open Knowledge Graph Embeddings? A Benchmark for Open Link Prediction. In *ACL*.

[4] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*.

[5] Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2020. A Simple Approach to Case-Based Reasoning in Knowledge Bases. In *AKBC*.

[6] Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM* (2015).

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

[8] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive Graph for Multi-Hop Reading Comprehension at Scale. In *PACL*.

[9] Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, et al. 2020. AutoKnow: Self-driving knowledge collection for products of thousands of types. In *SIGKDD*.

[10] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*.

[11] Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2018. OPIEC: An Open Information Extraction Corpus. In *AKBC*.

[12] Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open Knowledge Graph Embeddings. In *EMNLP*.

[13] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *SIGKDD*.

[14] Brett K Hayes, Evan Heit, and Haruka Swendsen. 2010. Inductive reasoning. *Wiley interdisciplinary reviews: Cognitive science* (2010).

[15] Gaole He, Junyi Li, Wayne Xin Zhao, Peiju Liu, and Ji-Rong Wen. 2020. Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning. In *WWW*.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*.

[17] Hassan Khosravi, Oliver Schulte, Tong Man, Xiaoyuan Xu, and Bahareh Bina. 2010. Structure learning for Markov logic networks with many descriptive attributes. In *AAAI*.

[18] Ni Lao, Tom Mitchell, and William Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *EMNLP*.

[19] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the Variance of the Adaptive Learning Rate and Beyond. In *ICLR*.

[20] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38 (1995).

[21] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In *ACL*.

[22] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *AAAI*.

[23] Jan Nuyts and Eric Pederson. 1999. *Language and conceptualization.*

[24] Raymond Reiter. 1981. On closed world data bases. In *Readings in artificial intelligence.* 119–140.

[25] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *ACL*.

[26] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference.*

[27] Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. 2019. An open-world extension to knowledge graph completion models. In *AAAI*.

[28] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *AAAI*.

[29] Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical Multi-Label Text Classification Using Only Class Names. In *NAACL*.

[30] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW*.

[31] Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *AAAI*.

[32] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge.

[33] Keith Stenning and Michiel Van Lambalgen. 2012. *Human reasoning and cognitive science.*

[34] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.

[35] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.

[36] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *ICLR*.

[37] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* (2014).

[38] Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin. 2018. Towards practical open knowledge base canonicalization. In *CIKM*.

[39] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*.

[40] Yuejia Xiang, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Zhenxi Lin, and Yefeng Zheng. 2021. OntoEA: Ontology-guided Entity Alignment via Joint Knowledge Graph Embedding. In *ACL-IJCNLP Findings*.

[41] Huiru Xiao and Yangqiu Song. 2021. Manifold Alignment across Geometric Spaces for Knowledge Base Representation Learning. In *AKBC*.

[42] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

[43] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment.. In *IJCAI*.

[44] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *AAAI*.

APPENDIX

A CREATION DETAILS AND STATISTICS OF TAXOKG-BENCH

Figures 2a-2f show the concept, entity and relation frequency histograms on six aligned TaxoKGs, where x-axis tick "*#m-n*" denotes the frequency bins ranges from m to n, and y-axis denotes the proportion of cases that falls into each bin. "* × ReVerb" in Figure 2 captions indicates that histograms are produced on the three AutoTAXOs aligned with the particular OpenKG constructed from *ReVerb*. Similarly, "* × ReVerb" indicates that histograms are produced on the three AutoTAXOS aligned with *OPIEC*. *MSCG* × *ReVerb* and *MSCG* × *OPIEC* are two large-scale TaxoKGs containing billions knowledge triplets of before filtering. Therefore, we set high thresholds for them. In particular, concepts with at least 20 grounded entities are kept in both MSCG × ReVerb and *MSCG* × *OPIEC* datasets, while entities with frequency greater than or equal to 40, 25 are kept in *MSCG* × *ReVerb* and *MSCG* × *OPIEC*, respectively. For relation, frequencies greater than or equal to 35, 3 are kept. Nevertheless, the remaining knowledge triplets are still in million scales, which makes the evaluation on these two *Taxo-KG*s very slow. We then conduct further down-samplings to build lightweight yet diverse testbeds. Similarly, we set the concept threshold, entity threshold, relation threshold for *SEMedical* aligned and *SEMusic* aligned *Taxo-KG*s as $\{3, 2, 2\}$ and $\{3, 4, 3\}$, respectively.

In Table 5, the columns *#entity*, *#concept* and *#pair* denote number of unique entities, concepts and entity-concept pairs reside in AutoTAXO part, while *#mention*, *#relation* and *#triplet* denote number of subject/object mentions, relation and subject-relation-object triplets reside in OpenKG part.

B SUMMARY OF EXISTING RELATION-GCN MODELS

The message passing functions of existing relational-GCN models can be viewed in Table 7. $\boldsymbol{h_u}, \boldsymbol{h_r}, \boldsymbol{h_v}$ denotes embeddings of source node u, relation r and target node v (message receiver). $\boldsymbol{W}$, $\boldsymbol{W_r}$, $\boldsymbol{W_{dir(r)}}$ denotes learnable weight matrices for all relations, each relation and each relation directions. $\boldsymbol{W_s}$ is a learnable weight matrix for self-loop edges. $\boldsymbol{\alpha_r}$ is a learnable weight scalar for each relation. For KBGAT, $[\cdot \parallel \cdot]$ denote vector concatenation operation. For CompGCN, ϕ is defined as composition operators.

C IMPLEMENTATION DETAILS

atan2 function. The atan2 function used § 5 in Eq. (7) is defined as follows:

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & if\ x > 0, \\ \arctan(\frac{y}{x}) + \pi & if\ x < 0\ and\ y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & if\ x < 0\ and\ y < 0, \\ \frac{\pi}{2} & if\ x = 0\ and\ y > 0, \\ -\frac{\pi}{2} & if\ x = 0\ and\ y < 0, \\ 0 & if\ x = 0\ and\ y = 0. \end{cases} \tag{13}$$

LtCaG model. Learn-to-Conceptualize-and-Generalize (LtCaG) model is a non-parametric model serving as one baseline method for TaxoKG completion task. The inference process of LtCaG can be explicitly depicted in the following equations:

$$\begin{aligned} &\mathcal{P}(< dog, CapableOf, bark >) = \\ &\quad 0.5 * \frac{\sum_v \mathcal{P}(< v, isA, dog >)\mathcal{P}(< v, CapableOf, bark >)}{\sum_v \mathcal{P}(< v, isA, dog >)} \\ &\quad + 0.5 * \frac{\sum_v \mathcal{P}(< dog, isA, v >)\mathcal{P}(< v, CapableOf, bark >)}{\sum_v \mathcal{P}(< dog, isA, v >)} \end{aligned} \tag{14}$$

$$\begin{aligned} &\mathcal{P}(< papillon, isA, dog >) = \\ &\quad 0.5 * \frac{\sum_{e,v} \mathcal{P}(< papillon, e, v >)\mathcal{P}(< dog, e, v >)}{\sum_{e,v} 1 - (1 - \mathcal{P}(< papillon, e, v >))(1 - \mathcal{P}(< dog, e, v >))} \\ &\quad + 0.5 * \frac{\sum_{e,v} \mathcal{P}(< v, e, papillon >)\mathcal{P}(< v, e, dog >)}{\sum_{e,v} 1 - (1 - \mathcal{P}(< v, e, papillon >))(1 - \mathcal{P}(< v, e, dog >))} \end{aligned} \tag{15}$$

Hyperparameters for Baselines and HakeGCN We implement HakeGCN using PyTorch[3] and DGL[4]. For compared methods, implementations are either from original authors (HAKE[5], CompGCN[6]) or dedicated replication (TransE, DistMult, HolE, R-GCN). We optimize HakeGCN and baselines through the Adam or RAdam [19] optimizer with learning rate $lr \in \{$1e-3, 3e-4, 1e-4$\}$ chosen by hyperparameter tuning on validation sets. For regularization, we choose an $l2$ penalty on all learnable parameters except PReLU layers and bias in fully-connected layers, with weights $C_{l2} \in \{0,$ 5e-5$\}$. Other hyperparameters include: token embedding size ($\{200, 300, 500\}$), entity and relation embedding size ($\{200, 500, 600, 800, 1000\}$), dropout ratio ($\{0.1, 0.3, 0.5\}$), negative sampling size ($\{1, 8, 32, 64, 128, 256\}$), batch size ($\{128, 256, 512, 1024\}$), epoch size ($\{200, 400, 800, 1200\}$).

For HakeGCN specific hyperparameters, we select the margin γ in Eq. (11) from $\{5, 8, 9, 10, 12\}$, and the temperature α in Eq. (12) from $\{0.5, 1.0, 1.5\}$. We use 2 GCN layers for the GCN encoder module to balance more high-order evidence with the over-smoothing issue, and we set all GCN layers embedding sizes the same as the entity and relation embedding sizes. For the taxonomy-based neighborhood sampling, the weight of keeping neighbors on the taxonomy is chosen from $\{1, 1.5, 2.0, 3.0, 5.0\}$, while the weight of keeping other neighbors is 1.

D CASE STUDIES OF NEIGHBORS FOR PREDICTING CONCEPTS AND RELATIONS

To have a more intuitive sense about the mutual enhancement of taxonomy and KG for TaxoKG completion, we show some examples of the neighbors used by HakeGCN in the AutoTAXO concept prediction task (Table 8) and the OpenKG relation prediction task (Table 9), where the check mark "✓", the question mark "?", or the cross mark indicate neighbors are beneficial, neutral or harmful for the prediction task, and "-" denotes concept itself serving as subject or object in the corresponding KG triplet. As we expect, neighbors from the taxonomy are mostly helpful for predicting the KG relations, and vice versa. For instance, when predicting

[3]PyTorch: https://pytorch.org/
[4]DGL: https://www.dgl.ai/
[5]HAKE: https://github.com/MIRALab-USTC/KGE-HAKE
[6]CompGCN: https://github.com/malllabiisc/CompGCN

Table 5: Statistics of the six datasets in TaxoKG-Bench.

Dataset	# entity	# concept	# pair	# mention	# predicate	# triplet
MSCG × ReVerb	5.6/1.0/3.6(K)	1.8/0.5/1.4(K)	6.4/1.2/4.0(K)	12.8/3.8/7.0(K)	10.3/2.2/4.8(K)	59.7/3.7/11.2(K)
SEMedical × ReVerb	256/48/163	261/131/219	256/48/163	7.3/1.3/2.9(K)	6.1/0.9/2.3(K)	21.3/1.3/4.0(K)
SEMusic × ReVerb	412/76/262	335/229/283	412/76/262	7.5/2.1/4.1(K)	8.9/1.7/3.7(K)	41.2/2.6/7.7(K)
MSCG × OPIEC	6.3/1.1/4.0(K)	1.8/0.6/1.4(K)	7.6/1.4/4.8(K)	5.5/1.8/3.2(K)	3.2/0.4/0.9(K)	51.2/3.2/9.6(K)
SEMedical × OPIEC	238/44/151	256/136/209	238/44/151	1432/255/564	508/75/199	2239/176/499
SEMusic × OPIEC	443/81/282	363/256/305	443/82/282	3.6/1.2/2.3(K)	1.4/0.3/0.6(K)	15.9/1.5/3.9(K)

Table 6: Percentages of unseen entities, concepts and relations in the testing set of the six datasets.

Dataset	Unseen Ent	Unseen Cept	Unseen Rel
MSCG × ReVerb	24.7%	39.6%	8.8%
SEMedical × ReVerb	14.4%	11.4%	15.5%
SEMusic × ReVerb	3.6%	3.2%	11.4%
MSCG × OPIEC	47.3%	30.0%	39.8%
SEMedical × OPIEC	18.1%	9.6%	15.1%
SEMusic × OPIEC	4.0%	0.7%	6.0%

Table 7: Summary of message passing functions in existing relational-GCN models.

Model	Message Passing Function
R-GCN [26]	$\boldsymbol{W_r h_u + W_s h_v}$
KBGAT [21]	$\boldsymbol{W[h_v \parallel h_u \parallel h_r]}$
SCAN [28]	$\boldsymbol{W \alpha_r h_u + W_s h_v}$
VR-GCN [43]	$\boldsymbol{W((h_v - h_r) + (h_u + h_r))}$
CompGCN [36]	$\boldsymbol{W_{dir(r)} \phi(h_u, h_r)}$

Table 8: KG neighbors used in taxonomy concept prediction.

Concept	KG Neighbors
technique	(make from, recycled material, -) ✓
	(architecture, be a thing of, -) ✓
	(-, be apply, biology) ✓
	(-, mean of, expression) ✗
disease	(-, have reach, epidemic proportion) ✓
	(-, can be treat in, a number of way) ✓
	(two, die of, -) ✓
	(alcohol, can cause, -) ✓
rock music	(-, be about, attitude) ✓
	(-, will start, a new era) ?
	(-, be a style of, music) ?
	(videos, recently tag with, -) ✗

Table 9: Taxonomy neighbors used in KG relation prediction.

Relation	Taxonomy Neighbors
be marry to	control ✗, family name ✓, guest ?
die from	illness ✓, disease ✓, disorder ✓
listen to	work of art ?, musical work ✓, piece of music ✓

the concept *disease* in Table 8, neighbors *(-, have reach, epidemic proportion)*, *(two, die of, -)*, and *(alcohol, can cause, -)* are supporting the correct prediction, although the neighbor *(-, can be treat in, a number of way)* may introduce some confusing evidence. On the other side, concepts *illness, disease, disorder* are helpful for predicting the relation *die from*. Therefore, the case studies clearly support our key insight about the mutual enhancement, and they shed a light on the future direction to investigate how to distinguish helpful and harmful neighbors towards further enhanced TaxoKG completion.

E EFFICIENCY EVALUATION

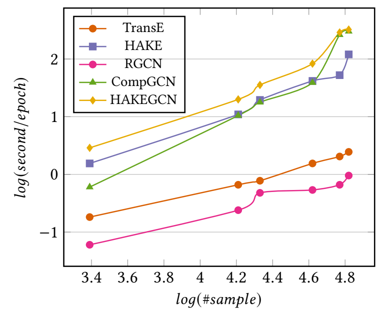


Figure 6: Model efficiency comparison in log scale.

We implement HakeGCN and all compared methods in Python and execute them on a server with two 48 cores Intel Xeon CPUs (768GB RAM), using one NVIDIA GeForce GTX 1080 Ti GPU (each with 24GB RAM). Figure 6 shows the runtimes of different models under various training sample sizes. HakeGCN shares similar time-complexity with HAKE and CompGCN. Although TransE and RGCN are more efficient, their performances are far from satisfactory. The slight extra time cost of HakeGCN is introduced by neighbor information aggregation and population, polar coordinate projection, and graph sampling.