Open-World Taxonomy and Knowledge Graph Co-Learning

Jiaying Lu JIAYING.LU@EMROY.EDU
Department of Computer Science
Emory University

Carl Yang J.CARLYANG@EMORY.EDU
Department of Computer Science
Emory University

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 integrate new knowledge. To empower KBs towards easy accommodation of emerging entities and relations, we propose to create open-world TaxoKGs 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. 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.

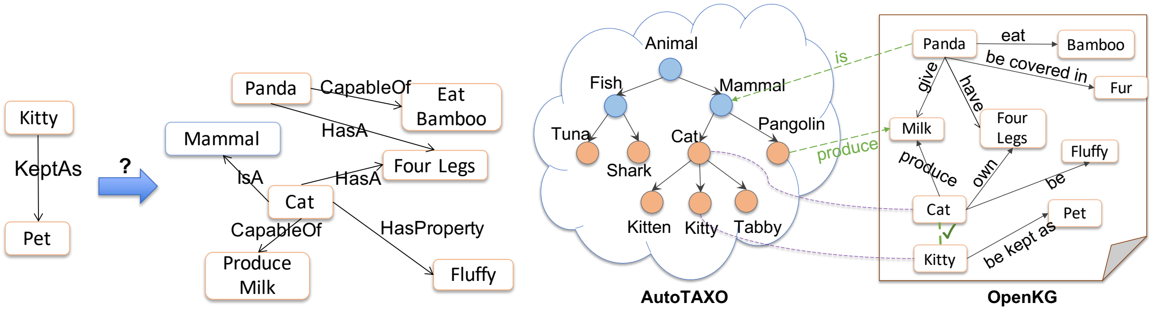
1. Introduction

Knowledge bases (KBs) have incorporated large-scale multi-relational data and motivated many knowledge-driven applications such as online encyclopedia [Vrandečić and Krötzsch, 2014] and e-commerce product catalog [Dong et al., 2020]. 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 stored in *taxonomies* (*e.g.*, "*(Cat, IsA, Mammal)*" in Fig. 1a);

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

Taxonomies are useful tools to organize and index concepts of entities so that users can efficiently find the information of interest [Shen et al., 2021, Mao et al., 2020]. On the other hand, KGs store human understanding of entities' properties, facts, or behaviors in a structured way, which are essential for knowledge representation and reasoning [Ding et al., 2019]. Extensive efforts have been made to construct KBs [Bollacker et al., 2008, Suchanek et al., 2007] that include both taxonomies and KGs. However, most existing KBs are in closed domains, and the creation process highly relies on pre-defined schema [Riedel et al.,



(a) Closed-world KB

(b) TaxoKG

Figure 1: Toy examples of existing KBs and TaxoKG.

2013] and exhaustive entity/relation canonicalization [Wu et al., 2018]. Although with guaranteed precision, closed-world KBs are limited in coverage and freshness. For example, if a KB is defined with a curated evolutionary biology schema that focuses on taxon and related characteristics of organisms, it is hard to incorporate knowledge triplets such as "(*Cat*, *KeptAs*, *Pet*) and (*German Shepherd*, *TrainedAs*, *Detection Dog*)". On the other hand, when a new triplet "(*Kitty*, *KeptAs*, *Pet*)" is introduced, although as humans we know kitty is a synonym of cat, the closed-world KB cannot easily incorporate the new knowledge unless the canonicalization tool can identify *Kitty* as *Cat*. Therefore, closed-world KB is most suitable for fixed or slowly evolving knowledge-enhanced applications.

Real-life applications often need to evolve with the fast-expanding entities and relations. To accommodate new emerging data, we propose to build open-world KBs with both taxonomic and non-taxonomic relations, namely Taxonomic Knowledge Graph (TaxoKG), by integrating automatically constructed taxonomies (AutoTAXOs) and open knowledge graphs (OpenKGs). An AutoTAXO is a collection of entity-concept pairs mined from web pages and search logs [Speer et al., 2017, Wu et al., 2012], and an OpenKG stores numerous factual triplets collected by open information extraction techniques from unstructured texts [Fader et al., 2011, Gashteovski et al., 2018]. Fig. 1b shows a toy example of TaxoKG. TaxoKG does not require curated schema or ad-hoc canonicalization. Instead, the adopted open-world setting empowers it to expand with new knowledge, and the integrated taxonomy serves as a soft schema for KG to mitigate the reliance on canonicalization (*e.g.*, *Kitten, Kitty* are children of *Cat*).

To understand the utility of TaxoKG, we create TaxoKG-Bench, a new benchmark with six datasets covering knowledge in general, medical, and music domains. To the best of our knowledge, this is the first effort on the integration of open-world 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 effective TaxoKG completion methods.

One significant challenge for open-world TaxoKG completion is to handle unseen entities, concepts and relations. Previous KB completion methods often rely on KB embeddings to predict the validity score of missing links [Bordes et al., 2013, Vashishth et al., 2020, Zhang et al., 2020]. However, these methods need to re-generate extra embeddings when presented with new data. Another key insight to complete TaxoKG is to leverage the mutual enhancement between taxonomies and KGs. Taxonomies convey rich context on inferring entities' properties and behaviors (*i.e.*, non-taxonomic relations). For example, humans have the common sense of "*mammal can produce milk*". Hence, if we encounter a rare mammal "*pangolin*", we can infer that "*pangolin can produce milk*". This reasoning ability is called "generalization" in cognition science [Hayes et al., 2010, Stenning and Van Lambalgen, 2012]. Furthermore, KGs are helpful for deducing entities' abstract concepts (*i.e.*, taxonomic relations). If we know "*mammal can eat and produce milk*", and "*pangolin can eat and produce milk*", it is highly possible that "*pangolin is a mammal*". This *conceptualize* ability is heavily used in information compression and human communication [Nuyts and Pederson, 1999, Davis and Marcus, 2015]. Existing KB completion methods, unfortunately, are not designed to leverage such mutual enhancement between taxonomies and KGs, 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 taxonomies and KGs. HakeGCN employs the polar coordinate embedding space to model the semantic hierarchy, and GCNs-based KB embeddings, to capture the higher-order organization. Moreover, HakeGCN creates entity, concept, and relation representations from their surface mentions, thus handling open-world challenges We examine HakeGCN and existing models on the TaxoKG completion task using the classical metrics and human evaluations. Then we conduct extensive ablation studies to evaluate (1) the utility of our technical designs; (2) the benefits of jointly modeling existing AutoTAXOs and KGs; (3) the impact of information propagation from taxonomic and non-taxonomic neighbors. We further present case studies for inferred knowledge and analyze the efficiency and scalability of HakeGCN in the Appendix.

2. Related Work

Knowledge Base Completion. Real-life KBs are usually incomplete [Dong et al., 2014]. KB completion aims at inferring missing facts based on the known facts. One popular approach is to embed entities and relations into vector spaces, and define a score function such that valid triples are assigned higher scores than the invalid ones. These KB embedding methods can be categorized into translation-based [Bordes et al., 2013, Sun et al., 2018, Zhang et al., 2020], semantic matching-based [Yang et al., 2015, Nickel et al., 2016], and neural network-based [Dettmers et al., 2018, Schlichtkrull et al., 2018, Vashishth et al., 2020]. More recently, HAKE [Zhang et al., 2020], inspired by RotatE [Sun et al., 2018], utilizes the modulus and phase information to model hierarchical relations. On the other hand, RGCN [Schlichtkrull et al., 2018] and CompGCN [Vashishth et al., 2020] incorporate graph neural network (GNN) as encoder to propagate the relation-specific information among 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 [Reiter, 1981] 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 [Shi and Weninger, 2018]. It is essential to infer knowledge about entities and relations not present in the existing KB, which is known as open-world KB completion [Gupta et al., 2019, Shah et al., 2019, Broscheit et al., 2020]. CaRe [Gupta et al., 2019] proposes a canonicalization-infused representation model to enrich OpenKB embeddings with the output of a canonicalization model, whereas OWE [Shah et al., 2019] predicts facts for unseen entities based on their textual descriptions.

Co-Learning of Taxonomy and Knowledge Graph. Previous works on taxonomies mainly focus on their automatic construction [Shen et al., 2020, Mao et al., 2020] and downstream tasks [Xiang et al., 2021, Shen et al., 2021]. On the other hand, extensive efforts have been put on KG construction [Suchanek et al., 2007, Bollacker et al., 2008], and KG-enhanced applications [Ding et al., 2019, He et al., 2020]. Although there exist attempts to collect closed-world KBs that contain both taxonomies and KGs [Miller, 1995, Bollacker et al., 2008, Suchanek et al., 2007, Speer et al., 2017], the two in the open-world setting (AutoTAXO and OpenKG) have rarely been studied together. JOIE [Hao et al., 2019] proposes a universal representation of entities and concepts for a two-view KB, which contains the ontology-view and the instance-view. GeoAlign [Xiao and Song, 2021] utilizes the manifold-aligned hyperbolic embedding for taxonomy and Euclidean embedding for KG to tackle the representation learning problem. Both JOIE and GeoAlign are designed for closed-world setting, thus not directly applicable to the open-world problems.

3. Problem Definition

The TaxoKG completion task is a variant of the general open-world KB completion:

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 triplets $\{(s,r,o)|(s,r,o) \notin \mathcal{E}, s \in \mathcal{V}^s, r \in \mathcal{R}^s, o \in \mathcal{V}^s\}$, where $\mathcal{V}^s$ and $\mathcal{R}^s$ are entity superset and relation superset, respectively.

More specifically, TaxoKG $\mathcal{B}$ contains the taxonomy $\mathcal{T}$ and the knowledge graph $\mathcal{G}$. An 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 taxonomic edges, all of which carry the uniform *IsA* relation. An 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 shared with $\mathcal{T}$, $\mathcal{R}_\mathcal{G}$ is the relation set that contains all other relations except for the taxonomic ones, 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. Hence, there exist two sub-tasks for TaxoKG completion: (1) the AutoTAXO concept prediction task and (2) the OpenKG relation prediction task. The former is to assign a set of concepts $C_e = \{c_1, c_2, \ldots, c_m\}$ for each entity $e \in \mathcal{V}_e$, whereas the latter 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 means we need to handle unseen entities, concepts, and relations.

4. TaxoKG-Bench: A New Benchmark with Six Datasets for TaxoKG

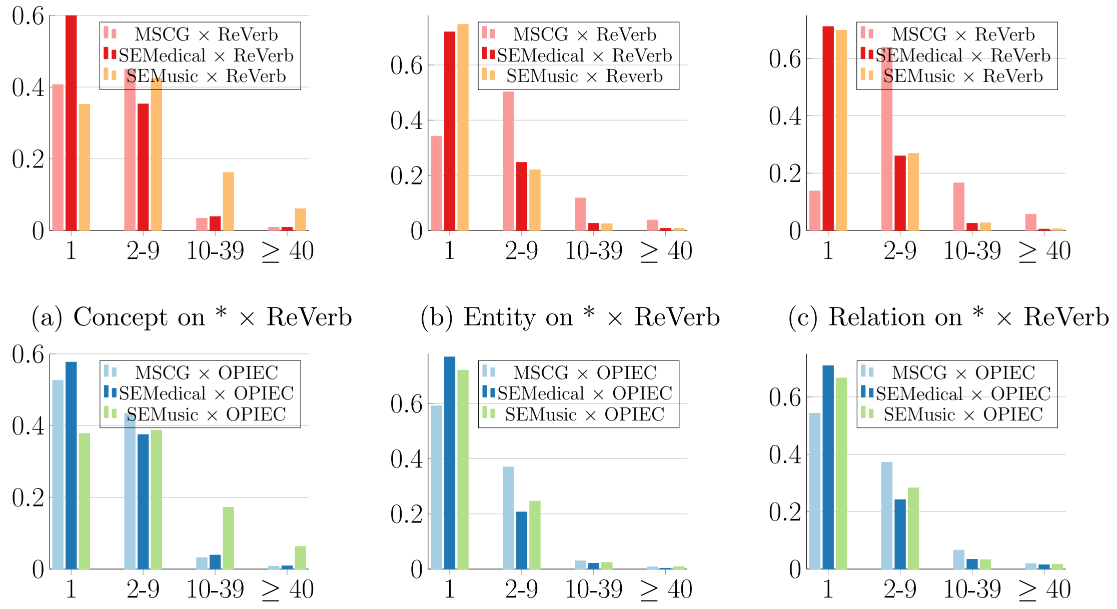
To the best of our knowledge, our work is the first to study the open-world taxonomy and knowledge graph co-learning problem. Hence, we create and release TaxoKG-Bench with six datasets of large-scale TaxoKG to the community for future studies[1].

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

4.1 Creation Process

The goal of building TaxoKG-Bench is to provide a benchmark to evaluate models on TaxoKG-based tasks such as its completion and applications. TaxoKG completion involves the ability to predict new-emerging concepts and novel facts for unseen entities. TaxoKG-Bench integrates the following data sources (details of them and the reason why we choose them are in Appx. A.1):

- Three AutoTAXOs: MS Concept Graph (MSCG) [Wu et al., 2012], SemEval-2018 Task 9 2A:Medical (SEMedical) and 2B:Music (SEMusic) [Camacho-Collados et al., 2018];
- Two OpenKGs: ReVerb [Fader et al., 2011] and OPIEC [Gashteovski et al., 2018].



(a) Concept on * × ReVerb (b) Entity on * × ReVerb (c) Relation on * × ReVerb

(d) Concept on * × OPIEC (e) Entity on * × OPIEC (f) Relation on * × OPIEC

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

To create TaxoKG-Bench, 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 to match entities. However, their reliabilities are far away from satisfactory. Alternatively, we use straight-forward string matching for the alignments. As can be seen from Fig. 2, there exist many long-tailed entities, concepts and relations, which makes the TaxoKG completion task more challenging. To make our benchmark efficient and easy to access for researchers, we conduct further post-processing (Appx. A.2) on the aligned six datasets to obtain the final released version.

4.2 Benchmark Overview

After the creation process mentioned above, we obtain the final version of six TaxoKGs, as can be seen in Table 4 (Appx. A.2). We aim to construct TaxoKG-Bench as a large-scale, diverse, challenging benchmark for TaxoKG completion. The coverage of TaxoKG-Bench is broad since the data sources come from general, medical, and music domains. The sizes range from thousands to hundred-thousands knowledge triplets. Moreover, the scale of relations is much larger than previous closed-world KBs. As an illustration, the widely-used closed-world KBs WN188RR [Miller, 1995] and FB15k-237 contains 11 and 237 relations separately, while our TaxoKG typically contains thousands of relations. Consequently, such large relation spaces requires stronger KB completion models. The proportions of taxonomic knowledge in TaxoKG-Bench vary from 1.4% to 13.5%, ensuring the benchmark's diversity. Among the great magnitude of entities, concepts, and relations, significant portions in the test set are never observed during training, as opposed to the assumption of closed-world KB. Table 5 (Appx. A.3) 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 unseen, which poses a serious challenge for the novel TaxoKG completion task.

5. HakeGCN: A Novel Method for Effective TaxoKG Completion

To tackle the TaxoKG completion task, our key insight is to leverage the mutual enhancement between taxonomy and KG. Hence, we propose a novel model with the learn-to-conceptualize and learn-to-generalize abilities via combining **H**ierarchy-**A**ware **K**nowledge base **E**mbedding and **G**raph **C**onvolutional neural **N**etworks, namely HakeGCN. As Fig. 5 (Appx. B.1) shows, it can be regarded as an encoder-decoder model. HakeGCN includes a series of essential technical designs for TaxoKG completion: (1) The polar coordinates-based GCN encoder (§ 5.2) which joins the power of GCNs in modeling multi-relations in KGs and polar coordinates in modeling hierarchical relations. (2) The taxonomy-based sampling strategy (§ 5.2) to improve the GCN encoder in learning from less-noisy neighbors. (3) The GCN-oriented phased bounded decoder (§ 5.3) that modify the value boundary of the original phase coordinate score function in HAKE [Zhang et al., 2020], making it easier for the decoder to differentiate entities at the same taxonomy level.

5.1 Handling Unseen Entities, Concepts and Relations

As §4.2 states, there are numerous unseen entities, concepts, and relations in the TaxoKG completion task. Unfortunately, most existing KB completion models [Bordes et al., 2013, Zhang et al., 2020, Schlichtkrull et al., 2018] are developed under the closed-world assumption, therefore their solution to embed entities/concepts/relations is to treat them as phrases and assign a look-up embedding table for phrases seen in the training set. Consequently, these models cannot handle new emerging phrases in the open-world setting. In HakeGCN, we opt to create entity, concept, and relation representations from the tokens of the surface mentions [Broscheit et al., 2020]. The entity and concept representations are then fed into the GCN encoder as initial embeddings of vertices $\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{0}}$, and relation representations as initial embeddings of edges $\boldsymbol{h}_{\boldsymbol{r}}^{\boldsymbol{0}}$. For more details please refer to Appx. B.2.

5.2 GCN Encoder with Polar Convolution and Taxo-based Neighbor Sampling

Updating Embeddings in Cartesian Coordinate. Since most existing KG embedding methods consider the input features or initial embeddings of entities and relations in the Cartesian coordinate system [Wang et al., 2017], we first adopt the widely-studied relational-GCNs in the Cartesian coordinate system. The choice of the GCN encoder (Appx. B.3) is flexible, as long as it takes both vertex and edge representations into account. We propose our own GCN encoder, which is a generalized form of existing relation-GCNs:

$$\boldsymbol{m_v^{k+1}} = \textsc{Agg}(\{\boldsymbol{W^k_{dir(r)}}\ \phi(\boldsymbol{h^k_u}, \boldsymbol{h^k_r}), \forall(u,r) \in \mathcal{N}(v)\}), \tag{1}$$

$$\boldsymbol{h_v^{k+1}} = \mathrm{PReLU}(\boldsymbol{W^k_v}[\boldsymbol{h^k_v} \parallel \boldsymbol{m^{k+1}_v}] + \boldsymbol{b^k_v}). \tag{2}$$

The message $\boldsymbol{m_v^{k+1}}$ on vertex v is collected from the neighbors $\mathcal{N}(v)$. 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 $\boldsymbol{h_u} \star \boldsymbol{h_r}$ [Nickel et al., 2016]. The aggregation operator $\textsc{Agg}(\cdot)$ can be chosen from *average*, *sum*, *max* or other functions. In practice, we select $\phi(\boldsymbol{h_u}, \boldsymbol{h_r})$ and $\textsc{Agg}(\cdot)$ through hyperparameter tuning. Moreover, the relation-specific learnable parameter $\boldsymbol{W_{dir(r)}}$ [Vashishth et al., 2020] in Eq. (1) is

$$\boldsymbol{W_{dir(r)}} = \begin{cases} \boldsymbol{W_o}, & (u,r,v) \in \mathcal{E}, \\ \boldsymbol{W_I}, & (u,r,v) \in \mathcal{E}_{inv}, \end{cases} \tag{3}$$

where $\mathcal{E}_{inv}$ denotes invert edges introduced to $\mathcal{B}$ for better vertex and edge representations. In Eq. (2), $[\boldsymbol{h^k_v} \parallel \boldsymbol{m^{k+1}_v}]$ denotes concatenation of the node and the message representations. Moreover, the edge updating rule is:

$$\boldsymbol{h_r^{k+1}} = \mathrm{PReLU}(\boldsymbol{W^k_r}\boldsymbol{h^k_r} + \boldsymbol{b^k_r}). \tag{4}$$

Mapping from Cartesian to Polar Representations. The polar coordinate-based embedding have shown promising results in closed-world KB completion [Sun et al., 2018, Zhang et al., 2020], as it utilizes the modulus dimension information to reflect depth of the taxonomy hierarchy and the phase dimension to represent the entities' surrounding non-taxonomic relations. To bridge the gap between the Cartesian coordinate embeddings from HakeGCN encoder and the polar coordinate embeddings used by decoder, we conduct the following representation mapping:

$$\rho = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \mathrm{atan2}(y, x), \tag{5}$$

where $x, y \in \mathbb{R}$, $\rho \in \mathbb{R}_+$, and $\theta \in [-\pi, +\pi]$. The atan2 function is a variation of the *arctangent* function (Appx. B.4). 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. (5), $\boldsymbol{h}$ can be mapped into $\boldsymbol{h} = [\boldsymbol{\rho} \parallel \boldsymbol{\theta}]$ containing d pairs of polar coordinates.

Taxonomy-based Neighborhood Sampling. We propose a taxonomy-based neighbor sampling strategy that intentionally keeps useful neighbors and discards noisy ones, which is an advancement of existing uniform neighbor sampling [Schlichtkrull et al., 2018]. The intuition is to allow the GCN encoder to see more neighbors close on the taxonomy, which contains less noise. The technical details are in Appx. B.5.

5.3 GCN-Oriented Phase Bounded Decoder

After getting the representations from the GCN-based encoder, the decoder scores "(subject, relation, object)" triplets through a function $f(s, r, o) : \mathbb{R}^d \times \mathbb{R}^{d'} \times \mathbb{R}^d \rightarrow \mathbb{R}$. We adopt the polar coordinate score function [Zhang et al., 2020] with a GCN-oriented boundary:

$$f(s, r, o) = -d(s, r, o) = -\lambda_m d_m(s, r, o) - \lambda_p d_p(s, r, o), \tag{6}$$

where (s, r, o) denotes both entity-concept pairs (with the associated relation of "IsA") and entity-relation-entity triplets in TaxoKG, and $d(s, r, o)$ denotes the distance function. In particular, $\lambda_m, \lambda_p \in \mathbb{R}$ are two learnable parameters to balance the modulus distance $d_m(s, r, o)$ and the phase distance $d_p(s, r, o)$. We also propose a GCN-oriented boundary for d_p for effective optimization. The technical details are elaborated in Appx. B.6.

Loss Function. We adopt the widely used negative sampling loss with self-adversarial strategy [Sun et al., 2018] for HakeGCN, of which the details are in Appx. B.7.

6. Experiments

In this section, we evaluate our proposed HakeGCN through performance comparisons, in-depth analysis, and ablation studies. Due to space limit, we present case studies (Appx. C.5), and efficiency evaluations (Appx. C.6) in the Appendix.

6.1 Experiment Settings

Evaluation Protocols. For the AutoTAXO concept prediction subtask of TaxoKG completion, we choose *Mean Average Precision* (MAP) and *Precision at N* (P@N) as evaluation metrics [Camacho-Collados et al., 2018]. MAP is based on top-15 predicted concepts. For the other OpenKG relation prediction subtask, we follow previous KB completion studies [Bordes et al., 2013] to rank candidate entities under the "filtered" protocol (Appx. C.1), and we choose *Mean Reciprocal Rank* (MRR) and *Hits at N* (H@N) as metrics.

Compared Methods. We adopt the following representative methods as baselines (for details please refer to Appx. C.2):

- Translation-based: TransE [Bordes et al., 2013], HAKE [Zhang et al., 2020];
- Semantic matching-based: DistMult [Yang et al., 2015], HolE [Nickel et al., 2016];
- GCN-based: R-GCN [Schlichtkrull et al., 2018], CompGCN [Vashishth et al., 2020];
- Mutual enhancement-based: LtCaG (Appx. C.2).

We integrate the same techniques introduced in §5.1 to mitigate unseen entities, concepts, and relations for baselines. The choices of hyperparameters are described in Appx. C.3.

6.2 Performance Comparisons

Tables 1a, 1b and 1c show the performance of compared models on TaxoKG-Bench. Our naïve LtCaG model, which requires no training, 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. Our HakeGCN consistently outperforms SOTA models on all datasets on both tasks, which demonstrates the substantial advantages

Table 1: TaxoKG completion results in different domains. For abbreviations, C-* indicates metrics for **c**oncept prediction, while R-* indicates metrics for **r**elation prediction. Underlined numbers denote the second runners, while bold numbers denote the winner.

(a) 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

(b) 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	O-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**

(c) 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**

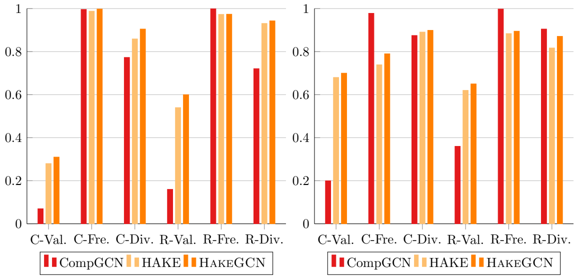
of integrating taxonomy and KG to mutually complete each other. There is an obvious pattern when entity and relation numbers grow from hundreds in SEMedical × OPIEC to ten-thousands in MSCG × ReVerb, where all baseline performances drop significantly due to the incapability of unseen entities and relations. HAKE is the second-best model that beats HakeGCN on some metrics in medical and general domains datasets. For more discussion please refer to Appx. C.4.

6.3 In-depth Analysis

We conduct in-depth analysis on knowledge triplets generated by two strongest baseline models CompGCN, HAKE, and HakeGCN, using the following manual (Validity) and automated (Freshness, Diversity) evaluating metrics:

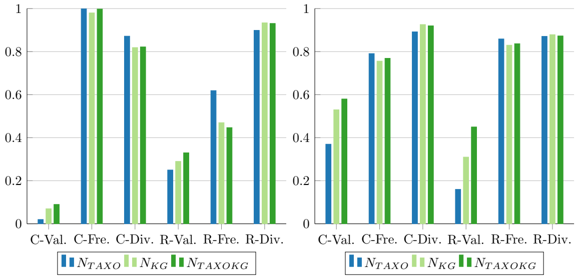
- Validity (Val.): whether generated triplets are valid to humans[2].

[2] The validity scores are annotated by two graduate students, Zishan Gu and Jiaying Lu, and three undergraduate students, Jacob Choi, Leisheng Yu, and Dheep Dalamal.



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

Figure 3: In-depth analysis for different models.



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

Figure 4: In-depth analysis for neighbors impact.

- Freshness (Fre.): the percentage of generated knowledge triplets that are novel[3].
- Diversity (Div.): Pielou's evenness index[4] which is popular in environment science to represent how equal the phrases in overall produced knowledge triplets 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. In Figure 3, the left three grouped bars (C-Val./Fresh./Div.) represent evaluation results of concepts assigned to entities of interest, and the right three stacked bars (R-Val./Fresh./Div.) represent results of generated open knowledge triplets. We observe that 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 Studies

Table 2: Ablation study results on HakeGCN technical designs.

	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

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 neighbor sampling (§5.2), polar GCN (§5.2), and GCN-oriented phase bounded decoder (§5.3). Table 2 presents the results on two medical TaxoKG's with the major metrics for both the AutoTAXO concept prediction (C-MAP) and the OpenKG relation prediction (R-MRR) tasks. For row "w/o. taxo_graph_sampling", we use the uniform neighbor sampling; for "w/o. polar_conv", we use the Cartesian coordinate-based graph convolution; for "w/o. phase_bounded_scorer", we use the existing unbounded

[3] A triplet not present in original TaxoKG is considered as fresh. Align with the open-world assumption, we treat each unique mention as a unique entity(concept, relation).

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

score function from HAKE. Table 2 supports the effectiveness of proposed techniques, since all three components improve the performance of HakeGCN.

Table 3: TaxoKG completion performance when presented with the separated data (SEMedical only or OPIEC only) v.s. the jointed data (SEMedical × OPIEC).

(a) Concept prediction results.

Model	Data	C-MAP	C-P@10, 30, 50
HAKE	AutoTaxo	.186	.344, **.355**, .177
HAKE	TaxoKG	**.262**	**.371**, .309, **.256**
CompGCN	AutoTaxo	**.075**	**.284**, **.117**, **.109**
CompGCN	TaxoKG	.041	.060, .044, .032
HakeGCN	AutoTaxo	.105	.093, .093, .123
HakeGCN	TaxoKG	**.271**	**.377**, **.366**, **.251**

(b) Relation prediction results.

Model	Data	R-MRR	R-H@10, 30, 50
HAKE	OKG	.350	**.454**, **.517**, **.545**
HAKE	TaxoKG	**.352**	.450, .509, .544
CompGCN	OKG	.006	.012, **.030**, **.049**
CompGCN	TaxoKG	**.009**	**.013**, .023, .034
HakeGCN	OKG	.375	.478, .555, .607
HakeGCN	TaxoKG	**.412**	**.508**, **.600**, **.652**

Can taxonomy and KG mutually enhance each other? To support the utility of TaxoKG integration, we further conduct ablation study on the taxonomy completion (concept prediction task) and KG completion (relation prediction task) performance when models are presented with only separated data instead of the jointed data of TaxoKG. The results clearly show the significant benefit of jointly modeling existing TAXOs and KGs. Specifically, our HakeGCN is the most effective one in leveraging such joined data of TaxoKGs (consistently achieving the most gains running on TaxoKGs over KGs and TAXOs only).

How do taxonomic and non-taxonomic neighbors impact the experiments? We further analyze the impact of neighbor information from AutoTAXOs and OpenKGs. In Figure 4, we plot the in-depth 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 3. As can be seen from Figure 4, 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 taxonomy and KG 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 the novel tasks of TaxoKG completion and application. Experiments on TaxoKG-Bench show that our novel KB completion model, HakeGCN, can effectively complete TaxoKG to further improve its coverage, so as to better support various knowledge-enhanced applications with rapidly evolving knowledge. In the future, it would be interesting to further integrate taxonomies and KGs with more sophisticated tools for TaxoKG creation.

References

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2013.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *ACL*, 2020.

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

Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. A simple approach to case-based reasoning in knowledge bases. In *AKBC*, 2020.

Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 2015.

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

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. Cognitive graph for multi-hop reading comprehension at scale. In *PACL*, 2019.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.

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. Autoknow: Self-driving knowledge collection for products of thousands of types. In *SIGKDD*, 2020.

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

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. Opiec: An open information extraction corpus. In *AKBC*, 2018.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. CaRe: Open knowledge graph embeddings. In *EMNLP*, 2019.

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

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

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

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

Hidetaka Kamigaito and Katsuhiko Hayashi. Comprehensive analysis of negative sampling in knowledge graph representation learning. In *ICML*, 2022.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, April 2020.

Yuning Mao, Tong Zhao, Andrey Kan, Chenwei Zhang, Xin Luna Dong, Christos Faloutsos, and Jiawei Han. Octet: Online catalog taxonomy enrichment with self-supervision. In *SIGKDD*, 2020.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38, 1995.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. In *ACL*, 2019.

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

Jan Nuyts and Eric Pederson. *Language and conceptualization*. 1999.

Raymond Reiter. On closed world data bases. In *Readings in artificial intelligence*, pages 119–140. 1981.

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

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

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

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

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

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. Taxoclass: Hierarchical multi-label text classification using only class names. In *NAACL*, 2021.

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

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. 2017.

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

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

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR*, 2018.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *ICLR*, 2020.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 2014.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *TKDE*, 2017.

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

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

Yuejia Xiang, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Zhenxi Lin, and Yefeng Zheng. Ontoea: Ontology-guided entity alignment via joint knowledge graph embedding. In *ACL-IJCNLP Findings*, 2021.

Huiru Xiao and Yangqiu Song. Manifold alignment across geometric spaces for knowledge base representation learning. In *AKBC*, 2021.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI*, 2019.

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

Appendix

Appendix A. More Details of TaxoKG-Bench

A.1 Introduction of Data Sources

We focus on the open-world setting where joining TAXOs and KGs can easily bring most benefits (because the open-world KBs themselves are less complete and canonicalized). Thus, we choose MSCG (general domain taxonomy), SEMedical (medical domain taxonomy), and SEMusic (musical domain taxonomy) because they are publicly available open-world taxonomies and relatively large. The reasons for choosing ReVerb and OPIEC are similar. Due to the open-world consideration, we do no use popular KBs such as YAGO [Suchanek et al., 2007], FreeBase [Bollacker et al., 2008] and ConceptNet [Speer et al., 2017]. Among them, MSCG is a large-scale AutoTAXO that contains millions of entity-concept pairs from billions of web pages, while SEMedical and SEMusic are two domain-specific 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.

A.2 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.

After the downsampling process mentioned above, we then split the six TaxoKGs into training, validation and testing sets for setting up a reproducible benchmark. On AutoTAXOs side, we split the entity-concept pairs by randomly assigning 55%, 5%, 35% entities into training, validation, testing set. On OpenKG side, we split subject-relation-object triplets by randomly assigning 80%, 5%, 15% triplets into training, validation, testing set.

Table 4: 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)

In other words, each split set is the union of assigned ontology-relation set and instance-relation set.

A.3 Unseen Concepts, Entities, and Relations in TaxoKG-Bench

Table 5: Percentages of unseen entities, concepts and relations in the testing set of the six datasets.

Dataset	Unseen Entity	Unseen Concept	Unseen Relation
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%

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 5 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. In Table 4, 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.

Appendix B. Technical Details of HakeGCN

B.1 HakeGCN Model Architecture

The overall architecture of HakeGCN is illustrated in Figure 5.

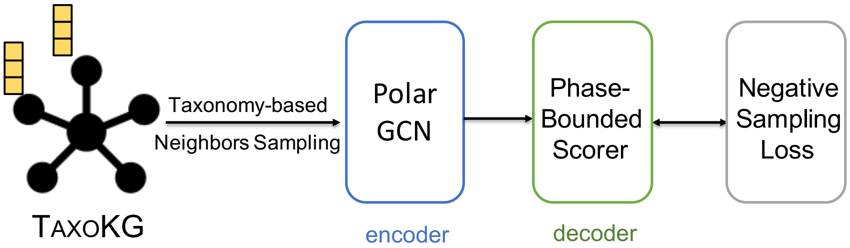


Figure 5: HakeGCN model architecture.

B.2 Obtaining Entity, Concept and Relation Representations in Open-World Setting

In HakeGCN, we opt to create entity, concept, and relation representation from the tokens of the surface mentions [Broscheit et al., 2020], to accommodate with new-emerging unseen phrases in the open-world setting. The entity and concept representations are then fed into the GCN encoder as initial embeddings of vertices $\boldsymbol{h_v^0}$, and relation representations as initial embeddings of edges $\boldsymbol{h_r^0}$. Therefore, for any vertex or edge h that is in the form of a sequence of tokens $\{t_1, t_2, \dots, t_L\}$, the representation is calculated by

$$\boldsymbol{h} = f(h) = f_{phr}(f_{tok}(t_1), f_{tok}(t_2), \dots, f_{tok}(t_L)), \tag{7}$$

where the lowercase letter h denotes vertex or edge phrase, the boldface lowercase letter $\boldsymbol{h}$ denotes the phrase embedding of vertex or edge, $f_{tok} : \mathbb{V}^{Tok} \rightarrow \mathbb{R}^d$ denotes the token embedding look-up mapping function, and $f_{phr} : \mathbb{R}^{L\times d} \rightarrow \mathbb{R}^{d'}$ 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:

$$\boldsymbol{h_v^0} = \text{PReLU}(\boldsymbol{W_v h_v} + \boldsymbol{b_v}) \quad \text{and} \quad \boldsymbol{h_r^0} = \text{PReLU}(\boldsymbol{W_r h_r} + \boldsymbol{b_r}). \tag{8}$$

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 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 [He et al., 2015]. The superscript 0 denotes that we use them as the input of the GCN encoder.

B.3 Summary of Existing Relational-GCN Models

The message passing functions of existing relational-GCN models can be viewed in Table 6. $\boldsymbol{h_u}, \boldsymbol{h_r}, \boldsymbol{h_v}$ 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.

Table 6: Summary of message passing functions in existing relational-GCN models.

Model	Message Passing Function
R-GCN [Schlichtkrull et al., 2018]	$\boldsymbol{W_r h_u} + \boldsymbol{W_s h_v}$
KBGAT [Nathani et al., 2019]	$\boldsymbol{W}[\boldsymbol{h_v} \parallel \boldsymbol{h_u} \parallel \boldsymbol{h_r}]$
SCAN [Shang et al., 2019]	$\boldsymbol{W \alpha_r h_u} + \boldsymbol{W_s h_v}$
VR-GCN [Ye et al., 2019]	$\boldsymbol{W}((\boldsymbol{h_v} - \boldsymbol{h_r}) + (\boldsymbol{h_u} + \boldsymbol{h_r}))$
CompGCN [Vashishth et al., 2020]	$\boldsymbol{W_{dir(r)}}\phi(\boldsymbol{h_u}, \boldsymbol{h_r})$

B.4 atan2 Function

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

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & if\ x > 0, \\ \arctan(\frac{y}{x}) + \pi & if\ x < 0\ and\ y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & if\ x < 0\ and\ y < 0, \\ \frac{\pi}{2} & if\ x = 0\ and\ y > 0, \\ -\frac{\pi}{2} & if\ x = 0\ and\ y < 0, \\ 0 & if\ x = 0\ and\ y = 0. \end{cases} \tag{9}$$

B.5 Taxonomy-based Neighborhood Sampling

Although the neighborhood information is helpful for KB completion tasks, many existing GCN-based models keep all neighbors during training which introduces noisy and even hazardous information [Ye et al., 2019, Vashishth et al., 2020]. 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, RGCN [Schlichtkrull et al., 2018] proposes to apply uniform random edge dropout on its encoder, which may discard useful neighborhood information. Therefore, we propose a taxonomy-based neighbor sampling strategy that intentionally keeps useful neighbors and discards noisy ones. Taxonomy-based sampling assigns a higher probability for edges between the entity of interest and the neighbors connected by both entity-entity 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 (Appx. C.3).

B.6 HakeGCN Decoder Score Function

Similar to HAKE [Zhang et al., 2020], the modulus and phase distance functions in Eq. (6) $f(s, r, o) = -\lambda_m d_m(s, r, o) - \lambda_p d_p(s, r, o)$, where (s, r, o) denotes both entity-concept pairs (with the associated relation of “IsA”) and entity-relation-entity triplets in TaxoKG, and $d(s, r, o)$ denotes the distance function. In particular, $\lambda_m, \lambda_p \in \mathbb{R}$ are two learnable parameters to balance the modulus distance $d_m(s, r, o)$ and the phase distance $d_p(s, r, o)$. The modulus and phase distance functions are defined by the following equations:

$$d_m(s,r,o) = \left\| \boldsymbol{h_{s,m}} \circ \boldsymbol{h_{r,m}} - \boldsymbol{h_{o,m}} \right\|_2 , \tag{10}$$

$$d_p(s,r,o) = \left\| \sin(\boldsymbol{h_{s,p}} + \boldsymbol{h_{r,p}} - \boldsymbol{h_{o,p}}) \right\|_1 , \tag{11}$$

where $\boldsymbol{h_s}, \boldsymbol{h_o}$ denote the subject, object embeddings obtained from the GCN encoder production $\boldsymbol{h_u}$ in Eq. (2), and $\boldsymbol{h_r}$ denotes the relation embedding obtained from a separate transformation in decoder using a similar process as in Eq. (4). For the polar coordinate, $\boldsymbol{h_{*,m}}, \boldsymbol{h_{*,p}}$ denote the embeddings in the modulus and phase part. In Eq. (10), the operator $\circ : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denotes the Hadamard product between two vectors. Let $\boldsymbol{\Delta\theta} = \boldsymbol{h}_{s,p} + \boldsymbol{h}_{r,p} - \boldsymbol{h}_{o,p}$. In the original phase distance function of HAKE, there is a denominator 2 for $\boldsymbol{\Delta\theta}$, which leads Eq. (11) to $\left\| \sin(\frac{\boldsymbol{\Delta\theta}}{2}) \right\|$. This is due to $\boldsymbol{h}_{*,p} \in [0, 2\pi)^d$, and thus $(\boldsymbol{h}_{s,p} + \boldsymbol{h}_{r,p} - \boldsymbol{h}_{o,p}) \in [0, 4\pi)^d$. 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. A similar idea can be seen in Sec. 3.2.1 of a recent paper [Kamigaito and Hayashi, 2022].

B.7 Negative Sampling Loss

We adopt the widely used negative sampling loss function [Bordes et al., 2013, Yang et al., 2015, Nickel et al., 2016, Zhang et al., 2020] with self-adversarial training [Sun et al., 2018]:

$$L = -\log \sigma(\gamma - d(s,r,o)) - \sum_{i=1}^{n} p(s'_i, r, o'_i) \log \sigma(d(s'_i, r, o'_i) - \gamma), \tag{12}$$

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:

$$p(s'_i, r, o'_i) = \frac{\exp(\alpha f_{samp}(s'_i, r, o'_i))}{\sum_j \exp(\alpha f_{samp}(s'_j, r, o'_j))}, \tag{13}$$

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

Appendix C. Detailed Experimental Settings and More Results

C.1 Introduction of "Filtered" Ranking Evaluation Protocol

We adopt the "filtered" evaluation protocol [Bordes et al., 2013], which is widely used in the general KG relation prediction problem, for our OpenKG relation prediction subtask. In OpenKG relation prediction, when predicting a triplet of interest, either subject or object is replaced with the candidate entity to create a set of candidate triplets. The candidate entities are then ranked in descending order, and ranking-based metrics are calculated over the ranked order. Unfortunately, these metrics, though might be indicative, can be flawed when some considered wrong triplets end up being valid ones. For instance, it is possible that there exist "<dog, capableOf, guard property>", "<dog, capableOf, be a pet>" and "<dog, capableOf, smell drugs>", in the training set; then when evaluating the triplet "<dog, capableOf, bark>" given query "<dog, capableOf, ?>", models may rank those triplets from training set above the test triplet. But this should not be counted as an error because both triplets are true. To avoid such misleading behavior, TransE authors [Bordes et al., 2013] propose to remove all the triplets that appear in the training, validation, or test set (except the test triplet of interest) from the candidate pool.

C.2 Introduction of Baseline Models

We adopt the following representative models as baselines:

- **Translation-based models** embed entities and relations into dense vector space, and define a score function such that valid triplets would be assigned higher scores than invalid ones.
  - TransE [Bordes et al., 2013] defines its score function for the triplet as $\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|$, where $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ denote the embeddings of head entity, relation and tail entity.
  - HAKE [Zhang et al., 2020] utilizes the polar coordinate to automatically learn the semantic hierarchy of entities without using clustering algorithms. Its score function is $\|\boldsymbol{h_m} \circ \boldsymbol{r_m} - \boldsymbol{t_m}\|_2 + \lambda \big\|\sin((\boldsymbol{h_p} + \boldsymbol{r_p} - \boldsymbol{t_p})/2)\big\|_1$, where the subscript $\boldsymbol{m}, \boldsymbol{p}$ denote the modulus part and phase part of HAKE polar embedding.
- **Semantic matching-based models** measure plausibility of triplets by matching latent semantics of embeddings of entities and relations.
  - DistMult [Yang et al., 2015] defines its score function as $\boldsymbol{h}^\top \boldsymbol{M_r} \boldsymbol{t}$, where $\boldsymbol{M_r}$ denotes a matrix for one relation which models pairwise interaction between head entity $\boldsymbol{h}$ and tail entity $\boldsymbol{t}$. DistMult restricts M_r to diagonal matrices to simplify the number of learnable parameters at the expense of inability to handle asymmetric relations.
  - HolE [Nickel et al., 2016] defines its score function as $\boldsymbol{r}^\top(\boldsymbol{h} \star \boldsymbol{t})$, where $\star$ denotes circular correlation operation that makes a compression of pairwise interactions between $\boldsymbol{h}$ and $\boldsymbol{t}$. Since the circular correlation operator $\star$ is not commutative, HolE can better handle asymmetric relations.
- **GCN-based models** incorporate powerful graph neural networks as encoders to propagate the relational information among interlinked entities, and utilize translational scoring function as decoder to infer the validity of edges.
  - R-GCN [Schlichtkrull et al., 2018] introduces relation-specific transformations in the neighbor information propagation. Hence, its message passing function is $\boldsymbol{W_r h_u} + \boldsymbol{W_s h_v}$, where $\boldsymbol{v}$ denotes target node, $\boldsymbol{u}$ denotes source node, $\boldsymbol{W_s}$ denotes the relation-specific transformation, and $\boldsymbol{W_s}$ denotes a self-connection transformation. R-GCN proposes basis and block-diagonal decomposition of relation specific filters for embeddings, which addresses the over-parameterization of vanilla relational GCNs.

- CompGCN [Vashishth et al., 2020] utilizes relation embedding $\boldsymbol{h_r}$ instead of the parameter matrix $\boldsymbol{W_r}$ to further avoid the over-parameterization issue. Its message passing function is defined as $\boldsymbol{W_{dir(r)}}\phi(\boldsymbol{h_u}, \boldsymbol{h_r})$, where $\boldsymbol{W_{dir(r)}}$ is a relation-direction specific parameter to distinguish whether it is a inbound, outbound or self-connection edge.

- **Mutual enhancement-based models** are designed to leverage the mutual enhancement between taxonomies and KGs.
  - LtCaG Learn-to-Conceptualize-and-Generalize (LtCaG) model is our own non-parametric model following the mutual enhancement intuition. LtCaG does not require any training process, which is similar to case-based reasoning model [Das et al., 2020]. Instead, the inference process is driven by the query triplet's prior and likelihood. The following equations explicitly depicted how LtCaG works. For instance, the probability of whether "<dog, capableOf, bark >" holds is determined by the probabilities of whether golden retrievers (subtypes of dog) or mammals (supertypes of dog) can bark. On the other hand, the probability of "<papillon, isA, dog>" is determined by the overlap between papillon's attributes and dog's attributes (non-taxonomic neighbors).

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

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

C.3 Hyperparameters for Baselines and HakeGCN

We implement HakeGCN using PyTorch[5] and DGL[6]. For compared methods, implementations are either from original authors (HAKE[7], CompGCN[8]) or dedicated replication (TransE, DistMult, HolE, R-GCN). We optimize HakeGCN and baselines through the Adam or RAdam [Liu et al., 2020] optimizer with learning rate $lr \in \{1\text{e-}3, 3\text{e-}4, 1\text{e-}4\}$ 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 $C_{l2} \in \{0, 5\text{e-}5\}$. Other hyperparameters include: token embedding size ($\{200, 300, 500\}$), entity and relation embedding size ($\{200, 500, 600, 800, 1000\}$),

[5] PyTorch: https://pytorch.org/

[6] DGL: https://www.dgl.ai/

[7] HAKE: https://github.com/MIRALab-USTC/KGE-HAKE

[8] CompGCN: https://github.com/malllabiisc/CompGCN

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. (12) from $\{5, 8, 9, 10, 12\}$, and the temperature α in Eq. (13) 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.

C.4 More Discussion on the Main TaxoKG Completion Experiments

Why does LtCaG achieve high performance? The baseline model LtCaG achieves high performance in the concept prediction task in medical and musical domain datasets, while performing not so well in the concept prediction task in general domain and all open KG completion tasks. LtCaG itself is a non-parametric model that reflects the intuition that "if a mammal can eat and produce milk, and if a pangolin can also eat and produce milk, then it is likely that a pangolin is a mammal" (for implementation details please refer to Appx. C.2). Both medical and musical domain datasets have small scales (less than 1,000) of concepts to predict, so LtCaG is very effective in these two domains. However, when presented with large amounts of concepts and entities to predict, LtCaG is significantly worse than parametric models.

Does HakeGCN beat other SOTA models? HakeGCN indeed outperforms SOTA models in almost every case. For instance, HakeGCN is consistently the best in the musical domain TaxoKGs (Tab. 1c) across all evaluation metrics; it is also the champion in medical domain TaxoKGs (Tab. 1b) across almost all (13/16) metrics except for C-P@3,10; HakeGCN also performs the best in general domain TaxoKGs (Tab. 1a) across 10/16 metrics. It achieves the second place in all cases if not the first, and it always leaves the remaining competitors far behind.

C.5 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 7) and the OpenKG relation prediction task (Table 8), 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 the concept *disease* in Table 7, 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 distinguish helpful and harmful neighbors towards further enhanced TaxoKG completion.

Table 7: 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 8: 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 ✓

C.6 Efficiency Evaluation

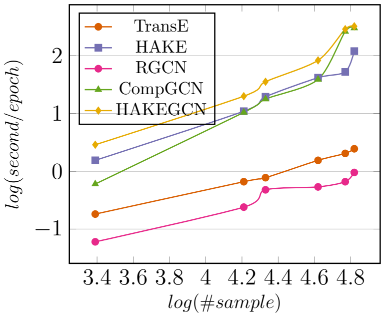


Figure 6: Model efficiency comparison in log scale.

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.