Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings

Erik B. Myklebust,a,b,f,* Ernesto Jiménez-Ruízc,b,** Jiaoyan Chend, Raoul Woltg and Knut Erik Tollefsena,e

a Norwegian Institute for Water Research, Oslo, Norway
b SIRIUS, University of Oslo, Oslo, Norway
c City, University of London, London, United Kingdom
d University of Oxford, Oxford, United Kingdom
e Norwegian University of Life Sciences, Ås, Norway
f NORSAR, Kjeller, Norway
g Norwegian Geotechnical Institute, Oslo, Norway

Abstract. We have created a knowledge graph based on major data sources used in ecotoxicological risk assessment. We have applied this knowledge graph to an important task in risk assessment, namely chemical effect prediction. We have evaluated nine knowledge graph embedding models from a selection of geometric, decomposition, and convolutional models on this prediction task. We show that using knowledge graph embeddings can increase the accuracy of effect prediction with neural networks. Furthermore, we have implemented a fine-tuning architecture which adapts the knowledge graph embeddings to the effect prediction task and leads to a better performance. Finally, we evaluate certain characteristics of the knowledge graph embedding models to shed light on the individual model performance.

Keywords: Knowledge graph, ecotoxicology, risk assessment, adverse effects, embedding, chemicals, species

1. Introduction

Ecotoxicology is a multidisciplinary field that studies the potentially adverse toxicological effects of chemicals on organisms, starting at molecular level to individuals, sub-populations, communities and ecosystems. One major societal contribution of ecotoxicology is ecological risk assessments, which compare environmental concentrations of chemicals with existing laboratory effect data to evaluate the ecosystem health status. While laboratory experiments are thus crucial, they are both labour intensive and result in a high number of animal testing. Therefore, the development of modelling techniques for extrapolating from existing laboratory effect data is a major effort in the field of ecotoxicology.

A very important challenge in ecotoxicology risk assessment is the interoperability of the disparate data sources, formats and vocabularies. The use of Semantic Web technologies and (RDF-based) knowledge graphs can address this challenge and facilitate the orchestration of these datasets. Hence, extrapolation or prediction models can benefit from an integrated view of the data and the background knowledge provided by a knowledge graph. The use of knowledge graphs also enables the use of the available infrastructure to perform automated reasoning, explore the data via se-
mantic queries, and compute semantic embeddings for machine learning prediction.

In this work we have created the Toxicological Effect and Risk Assessment Knowledge Graph (TERA) and implemented a prediction model over this knowledge graph to extrapolate adverse biological effects of chemicals on organisms. Here, we limit ourselves to binary effect prediction of mortality (shortened to effect prediction), i.e., where there is a chance that a chemical can affect a species in a lethal way. The work and evaluation conducted in this paper is driven by the following research question: does the use of contextual information in the form of knowledge graph embeddings brings added value in the prediction of adverse biological effects?

Our contributions can be summarized as follows:

(i) TERA aims at consolidating the relevant information to the ecological risk assessment domain. TERA integrates several disparate datasets and enables a unified (semantic) access. The formats of these data sources vary from tabular, to RDF files and SPARQL endpoints over public linked data. We have exploited external resources (e.g., Wikidata [76]) and ontology alignment methods (e.g., LogMap [34]) to discover equivalences between the data sources.

(ii) We have designed and implemented a model tailored to binary lethal chemical effect prediction. This model relies on TERA and builds upon existing knowledge graph embedding models. Moreover, it supplies the knowledge graph embedding models with additional information. This is used to tailor the embeddings to this specific task.

(iii) We have evaluated nine knowledge graph embedding (KGE) models, together with a naive baseline on the binary chemical effect prediction task. This evaluation includes four data sampling strategies which highlight the different settings of chemical effect prediction (i.e., the test data contains unseen chemical-organism pairs where: (a) the chemical and the organism may be known (but not in previously seen pairs), (b) the chemical is unknown, (c) the organism is unknown, and (d) both the chemical and the organism are unknown).

These contributions are openly shared. A snapshot of the TERA knowledge graph is available on Zenodo [52] (https://doi.org/10.5281/zenodo.3559865) and the source scripts for creating TERA are available on GitHub (https://github.com/NIVA-Knowledge-Graph/TERA). Finally, the scripts to reproduce the conducted evaluation in this paper are also available on GitHub (https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020).

This paper extends our preliminary work presented in the In-Use Track of the 18th International Semantic Web Conference [50]. We have (i) extended TERA with new sources (Encyclopedia of Life (EOL), MeSH, and a larger part of ChEMBL) and provided detailed steps about its creation; (ii) created a more robust prediction model with nine (up from three) embedding algorithms supported and a task-specific embedding fine-tuning strategy; and (iii) conducted a more comprehensive evaluation with all combinations of KGE models and sampling strategies totalling 648 data points (324 for each prediction model).

The rest of the paper is organized as follows. Section 2 introduces essential concepts to the subsequent sections. Section 3 introduces the use case where the knowledge graph and prediction models are applied. Section 4 introduces related work. The creation of the knowledge graph is described in Section 5. Section 6 introduces the prediction models, while Section 7 presents the evaluation of these models. Section 8 elaborates on the contributions and discusses future directions of research. Finally, Appendix A gives an overview of the knowledge graph embedding models used in this work.

2. Preliminaries

In this section we introduce important background concepts that will be used throughout the paper. Table 1 contains the most important symbols.

2.1. Ecotoxicological terminology

Taxonomy in this work refers to a species classification hierarchy. Any node in a taxonomy is called a taxon. Species is a taxon which is also a leaf node in the taxonomy. An Organism denotes an individual living organism which is an instance of a species. Chemicals or compounds are unique isotopes of substances consisting of two or more atoms. Effect, used in this work as short form for chemical effect, refers to the response of an organism (or population) to a chemical at a specific concentration. Endpoint[^1] denotes a measured ef-
Table 1

| Symbol | Definition |
|-------|------------|
| RDF   | Resource Description Framework |
| OWL   | Web Ontology Language |
| SPARQL| SPARQL Protocol and RDF Query Language |
| KG    | Knowledge graph |
| KGE   | Knowledge graph embedding |
| t     | A triple |
| sb    | The subject of a triple |
| ob    | The object of a triple |
| p,r   | The predicate/relation of a triple |
| e     | A KG entity |
| T     | The set of KG triples |
| E     | The set of KG entities |
| R     | The set of KG relations |
| L     | The set of literal values |
| e     | The vector representation of an entity or relation |
| k     | The dimension of a vector |
| SF    | The scoring function of a KGE model |
| PT    | Pre-trained KGE-based model |
| FT    | Fine-tuning KGE-based model |
| s     | A species |
| c     | A chemical |
| S     | Refers to species |
| C     | Refers to chemicals |
| κ     | Chemical concentration |

Table 1

Key symbols and acronyms used throughout the paper.

An (ontology-enhanced) KG can be split into a TBox (terminology) and an ABox (assertions). The TBox is composed by triples using RDF Schema (RDFS) constructors like class subsumptions and property domain and range; and OWL constructors like disjointness, equivalence and property inverses. The ABox contains assertions among instances, including OWL equality and inequality, and semantic type definitions. Table 5 shows several examples of TBox and ABox triples.

2.3. Ontology alignment

Ontology alignment is the process of finding mappings or correspondences between a source and a target ontology or knowledge graph [23, 66]. These mappings typically represent equivalences or broader/narrower relationships among the entities of the input ontologies. In the ontology matching community [61], mappings are exchanged using the RDF Alignment format [18]; but they can also be interpreted as standard OWL axioms (e.g., [24, 35]). In this work we treat ontology alignments as OWL axioms (e.g., Triple t13 in Table 5). An ontology matching system (e.g., LogMap [33]) is a program that, given as input two ontologies or knowledge graphs, generates as output a set of mappings (i.e., an alignment) M.

2.4. Embedding models

Knowledge graph embedding (KGE) [63, 78] plays a key role in link prediction problems where it is applied to knowledge graphs to resolve missing facts in largely connected knowledge graphs, such as DBpedia [44]. Biomedical link prediction is another area where embedding models have been applied successfully (e.g., [1, 5]).

Thus, in our setting, KGs are composed by RDF triples in the form of \((sb, p, ob) \in E \times R \times E \cup L\), where \(sb\) represents a subject (a class or an instance), \(p\) represents a predicate (a property) and \(ob\) represents an object (a class, an instance or a literal). KG entities (i.e., \(E \cup R\): classes, properties and instances) are represented by an URI (Uniform Resource Identifier).

Note that the Web Ontology Language (OWL) [27] also enables the creation of complex axioms that are translated/serialized into more than one triple: https://www.w3.org/TR/owl2-mapping-to-rdf/.

For the embedding process, we focus on triples where \(o \in E\) is a class or an instance.
Exposure
Effects
Risk Quotient

Risk
Hot spot identification

Susceptible Species

Fig. 1. Simplified ecological risk assessment pipeline.

$\mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$, $SF \propto P((sb, p, ob) \in KG)$; and (ii) minimizing a loss function (i.e., deviation of the prediction of the scoring function with respect to the truth available in the KG). More specifically, KGE models (i) initialize the entities in a triple $(sb, p, ob)$ into a vector representation $e_{sb}, e_{p}, e_{ob} \in \mathbb{R}^k$ or $\mathbb{C}^k$, where $k$ is the dimension of the vector; (ii) apply a scoring function to $(e_{sb}, e_{p}, e_{ob})$; and (iii) adapt the vector representations to improve the scoring and minimize the loss.

Several knowledge graph embedding models have been proposed. In this work, we used models of three major categories: decomposition models, geometric models, and convolutional models. The decomposition models represent the triples of the KG into a one-hot 3-order tensor and apply matrix decomposition to learn entity vectors. Geometric models, also known as translational, try to learn embeddings by defining a scoring function where the predicate in the triple act as a geometric translation (e.g., rotation) from subject to object. Convolutional models, unlike previous models, learn entity embedding with non-linear scoring functions via convolutional layers.

3. Ecotoxicological risk assessment and adverse biological effect prediction

The task of ecotoxicological risk assessment is to study the potential hazardous effects of chemicals on organisms from individuals to ecosystems. In this context, risk is the result of the intrinsic hazards of a substance on species, populations or ecosystems, combined with an estimate of the environmental exposure, i.e., the product of exposure and effect (hazard).

Figure 1 shows a simplified risk assessment pipeline. Exposure data is gathered from analysis of environmental concentrations of one or more chemicals, while effects (hazards) are characterized for a number of species in the laboratory as a proxy for more ecologically relevant organisms. These two data sources are used to calculate the so-called risk quotient (RQ: ratio between exposure and effects). The RQ for one chemical or the mixture of many chemicals is used to identify chemicals with the highest RQs (risk drivers), identify relevant modes of action (MoA) and characterize detailed toxicity mechanisms for one or more species (or taxa). Results from these predictions can generate a number of new hypotheses that can be investigated in the laboratory or studied in the environment. Note that, this risk assessment pipeline is a simplified version of the one in use at the Norwegian Institute for Water Research, however, similar methodologies are used across regulatory risk assessment pipelines.

The chemical effect data is gathered during laboratory experiments, where a sub-population of a single species is exposed to an increasing concentration of a toxic chemical. The endpoints of the experiments are recorded at chemical concentrations and time after exposure. These endpoints are categorized into several categories, e.g., lethality rate of test population (see Table 2).

Ecological risk assessment methods require a large amount of these experimental data to give an accurate depiction of the long term risk to an ecosystem. The data must cover the relevant chemicals and species present in the ecosystem, e.g., an ecological risk assessment of agricultural runoff in Norway will mostly concern pesticides and waterflees, copepods, and frogs, among other species [42]. Just with a few relevant chemicals and species the search space becomes immense and performing laboratory experiments becomes unfeasible. Thus, it is essential to develop in silico methods to extrapolate new chemical-species effects from known combinations. We differentiate among two types complementary strategies: (i) highly specialized (restricted in chemical and species domains) models to predict chemical concentrations that will have an effect on a test species, and (ii) models that produce rankings of highly representative chemical-species pair hypothesis which can be

---

6The interested reader please refer to [63] for a comprehensive survey.

7The mode of action describes the molecular pathway by which a chemical causes physiological change in an organism.

8NIVA: [https://www.niva.no/en](https://www.niva.no/en)
Table 2

| Endpoint   | Frequency | Description                                      |
|------------|-----------|--------------------------------------------------|
| NR         | 0.21      | Not reported                                     |
| NOEL       | 0.17      | No-observable-effect-level                       |
| LC50       | 0.16      | Lethal concentration for 50% of test population  |
| LOEL       | 0.14      | Lowest-observable-effect-level                   |
| NOEC       | 0.05      | No-observable-effect-concentration               |
| EC50       | 0.05      | Effective concentration for 50% of test population|
| LOEC       | 0.04      | Lowest observable effect concentration           |
| BCF        | 0.03      | Bioconcentration factor                          |
| NR-LETH    | 0.02      | Lethal to 100% of test population                |
| LD50       | 0.02      | Lethal dose for 50% of test population           |
| Other      | 0.11      |                                                  |

The most frequent endpoints in ECOTOX [74] chemical effect data.

used by a laboratory to perform targeted experiments. In this paper we focus on the latter strategy, using a method based on knowledge graph embeddings. Methods that fall into the first strategy are introduced in Section 4.1.

4. Related work

This section will cover related work from ecotoxicology and knowledge graph based prediction.

4.1. Toxicity extrapolation

There are two main research areas in toxicology to extrapolate chemical effects, i.e., Quantitative Structure-Activity Relationship (QSAR) and read-across. QSAR modelling try to find a relationship between the structure of a chemical and the chemical’s biological activity (cf. reviews [22, 26]). This relationship is described using derived chemical features. Some features are simple, e.g., octanol-water partition coefficient or logP, others concern the entire chemical, e.g., chemical fingerprints. The basis of the QSAR relationship is usually modeled as polynomial equations. Parthasarathi and Dhawan [59] take this further by using the logarithm of chemical concentration to achieve a polynomial relationship: \( \log(1/\kappa) = f(\pi) + g(\sigma) \), \( f \in P_n \) and \( g \in P_1 \) (\( P_n \) is a polynomial of \( n \)th degree), where \( \kappa \) is the chemical concentration while \( \pi \) and \( \sigma \) denote the derived chemical features hydrophobicity and electronic effects in the molecule, respectively. The drawback of these models is the applicability domains. Usually, a QSAR model considers a small set of chemicals (10ths to 100ths) and one single species. This means that new features and relationships need to be developed for each species and each chemical group.

The read-across methods try to mitigate these drawbacks, mainly by considering extrapolation of the effect at the chemical and species levels. Similar to QSAR models, read-across of chemicals use the chemical features to create similarity measures between chemicals to justify the read-across of chemical effects. The read-across in the species domain is harder. Species do not tend to have easily derived features. Therefore, genetic similarity has emerged as a viable option. Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS), developed by the United States Environmental Protection Agency (U.S. EPA.), is an example of such an approach [20, 41]. SeqAPASS uses a large amount of data available for humans, mice, rats, and zebrafish to extrapolate to areas with lower coverage.

4.2. Embedding models

In this work, we use nine KGE models across three categories of models. Here, we will give a brief introduction to the models, while a more extended explanation of the models is found in Appendix A. The interested reader please refer to [63] for a comprehensive survey.

The three categories of models are decomposition, geometric, and convolutional [63]. The decomposition models are DistMult, ComplEx, and HolE. DistMult

\(^9\)Measure of the absence of attraction to water.
models the score of a triple as the vector multiplication of the representation of each subject, predicate and object \[83\]. ComplEx uses the same scoring function as DistMult, however, in a complex vector space, such that it can handle inverse relations \[73\]. HolE is based on holographic embeddings \[55\], however, it has been shown that HolE is equivalent to ComplEx \[30\]. The geometric models are TransE, RotatE, pRotatE, and HAKE. TransE is the base of a whole family of models and scores triples based on the translation from subject to object using the representation of the predicate \[10\]. RotatE is similar to TransE, however, the translation using the predicate is done by rotating it (via Euler’s identity) \[70\]. Furthermore, pRotatE is a baseline for RotatE where the modulus in Euler’s identity is ignored \[70\]. Finally, the hierarchical-aware model, HAKE, where entities at each level in the hierarchy is at equal distance from the origin and relations at a level is modeled as rotation \[85\].

The convolutional models take a deep learning approach to the task of KGE. We use ConvKB \[54\] and ConvE \[19\], which are similar with slightly different architectures. They have shown good performance given the relative small number of parameters.

Although quite a few KGE models have been proposed, the adopted ones are either classic models or can achieve state-of-the-art performance in some benchmarks. They are representative of mainstream techniques, and have been widely adopted in KGE research and applications \[63\]. Thus, the benefits and shortcomings of the KGE models analysed in this study provide good evidence of the general performance of this type of models in a complex prediction task, i.e., adverse biological effect of chemicals on organisms.

4.3. Using KGE for prediction

Our focus to use KGE models is to predict if a chemical has a lethal effect on an organism. KGE models have been explored in the biomedical domain to solve similar predictions tasks (e.g., finding relationships between diseases, drugs, genes, and treatments). Several works have shown improvements in results by using KGE models for prediction, e.g., \[1\] \[5\] \[45\]. Chen et al. \[15\] used random walks over networks to perform drug-target predictions. The ChEMBL and DrugBank KGs have also been used to predict chemical mode of action (MoA) of anticancer drugs with high performance on benchmark datasets \[92\].

Opa2vec \[68\] and Blagac et al. \[8\] have developed embedding models to improve similarity-based prediction in the biomedical domain, while OpenBi-oLink \[12\] has created a framework for evaluating models in the biomedical domain.

EL Embeddings \[40\] and opa2vec \[68\] present new semantic embedding methods for KGs with expressive logic expressions (i.e., OWL ontologies) to predict protein interaction. The former utilizes complex geometric structures to model the logic relationships between entities, while the later learns a language model from a corpus extracted from the ontology. OWL2Vec* \[13\] also learns a language model from an ontology and applies the computed embeddings into two prediction tasks: class subsumption and class membership. OWL2Vec* has also been used to predict the plausibility of ontology alignments \[14\].

To the best of our knowledge there is no work using link prediction or KGE models to support ecotoxicological effect prediction. This study will give novel insights and empirical results of KGE models in this new domain.

5. TERA knowledge graph

One major challenge in ecological risk assessment processes is the interoperability of data. In this section, we introduce the Toxicological Effect and Risk Assessment (TERA), an ontology-enhanced RDF-based knowledge graph that aims at providing an integrated view of the relevant data sources for risk assessment \[10\].

The initial inspiration for TERA was the aid of ecotoxicological effect prediction where access to disparate resources was required (see Section 5.3). However, by integrating these sources into a KG, we were also able to directly apply TERA into the prediction process by leveraging knowledge graph embedding models (see Section 5.4).

The data sources integrated into TERA vary from tabular and RDF files to SPARQL endpoints over public linked data. The sources currently integrated into TERA are: (i) biological: NCBI Taxonomy, Encyclopedia of Life, and Wikidata mappings (~ 500k species); (ii) chemical: PubMed, ChEMBL, MeSH, and Wikidata mappings (~ 110M compounds); and (iii) biological effects: ECOTOXicology Knowledgebase (~ 1M results, ~ 12k compounds, ~ 13k species).

\[83\] Resources to create and access TERA: https://github.com/NIVA-Knowledge-Graph/TERA
Fig. 2. Data sources and processes to create the TERA knowledge graph.

and system-generated mappings. These three distinct parts make up the sub-KGs of TERA, i.e., (i) the Taxonomy sub-KG (KG₆), (ii) the Chemical sub-KG (KG₇), and (iii) the Effects sub-KG (KG₈). The different processes to transform and integrate these sources into TERA are shown in Figure 2.

A snapshot of TERA is available on Zenodo [52], where licenses permit [52]. PubChem and ChEMBL are not included in the snapshot due to size constraints; these can be downloaded from the National Institutes of Health [12] and European Bioinformatics Institute [13] respectively. The subgraph of TERA used for prediction is available alongside the chemical effect prediction models in our GitHub repository [14]. Table 5 shows several examples of RDF triples from TERA [15].

5.1. Dataset overview

TERA, as mentioned above, is constructed by gathering a number of sources about chemicals, species and chemical toxicity, with a diverse set of formats including tabular data, RDF dumps and SPARQL endpoints.

Biological effect data of chemicals. The largest publicly available repository of effect data is the ECO-TOXicology knowledgebase (ECOTOX) developed by the US Environmental Protection Agency [74]. This data is gathered from published toxicological studies and limited internal experiments. The dataset consists of 1M experiments covering 12k chemicals and 13k species implying a chemical–species pair converge.

compound: {PubChem compound}, descr: {PubChem descriptors}, vocab: {PubChem vocabulary}, inchkey: {InChIKey identifiers}, env: {Environment Ontology} cheminf: {Chemical information ontology}, chembl: {ChEMBL}, chembl_m: {ChEMBL molecule subset}, chembl_t: {ChEMBL target subset}, wd: {WikiData entities}, wdt: {WikiData properties}, qudt: {Quantities, Units, Dimensions and Types Catalog}, snomedct: {SNOMED CT ontology}, and bp: {Biological PAthway eXchange ontology}. owl:, rdfs:, rdf: and xsd: are prefixes referring to W3C standard vocabularies.

11EOL: Various Creative commons (CC), NCBI: Creative Commons CC0 1.0 Universal (CC0 1.0), ECOTOX: No restrictions, PubChem: Open Data Commons Open Database License, ChEMBL: CC Attribution, MeSH: Open, Courtesy of the U.S. National Library of Medicine, Wikidata: CC0 1.0.

12ftp://ftp.ncbi.nlm.nih.gov/pubchem/RDF/

13ftp://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBL-RDF/

14https://github.com/NIV A-Knowledge-Graph/KGs_and_Effect_Prediction_2020

15Prefixes associated to the URI namespaces of entities in TERA: et: (ECOTOXicology knowledgebase), ncbi: (NCBI taxonomy), eol: (Encyclopedia of Life), mesh: (Medical Subject Heading),

16Version dated Sep. 15, 2020.
of maximum $\sim 0.6\%$. The resulting endpoint from an experiment is categorised in one of a plethora of predefined endpoints (see Table 2 above).

Tables 3 and 4 contain an excerpt of the ECOTOX database. ECOTOX includes information about the chemicals and species used in the tests. This information, however, is limited and additional (external) resources are required to complement ECOTOX.

**Chemicals.** The ECOTOX database uses an identifier called CAS Registry Number assigned by the Chemical Abstracts Service to identify chemicals. The CAS numbers are proprietary, however, Wikidata \cite{76} (indirectly) encodes mappings between CAS numbers and open identifiers like InChIKey, a 27-character hash of the International Chemical Identifier (InChI) which encodes chemical information uniquely \cite{31}. Wikidata also provides mappings to well known databases like PubChem, ChEMBL and MeSH, which include relevant chemical information such as chemical structure, structural classification and functional classification.

**Taxonomy.** ECOTOX contains a taxonomy\footnote{In the context of the paper "taxonomy" typically refers to a classification of organisms.} of species, however, this only considers the species represented in the ECOTOX effect data. Hence, to enable extrapolation of effects across a larger taxonomic domain, we include the NCBI Taxonomy \cite{64}. This taxonomy data source consists of a number of database dump files, which contains a hierarchy for all sequenced species, which equates to around 10\% of the currently known life on Earth and is one of the most comprehensive taxonomic resources. For each of the taxa (species and classes), the taxonomy defines a handful of labels, the most commonly used of which are the scientific and common names. However, labels such as authority can be used to see the citation where the species was first mentioned, while synonym is an alternate scientific name, that may be used in the literature.

**Species traits.** As an analog to chemical features, we use species traits to expand the coverage of the knowledge graph. Apart from taxonomic classifications, traits are the most important information to identify species and will be of great importance when predicting the effect on the species.

The traits we have included in the knowledge graph are the habitat, endemic regions, and presence (and classifications of these). This data is gathered from the Encyclopedia of Life (EOL) \cite{57}, which is available as a property graph. Moreover, EOL uses external definitions of certain concepts, and mappings to these sources are available as glossary files. In addition to traits, researchers may be interested in species that have different conservation statuses, e.g., if the population is stable or declining, etc. This data can also be extracted from EOL.

### 5.2. Dataset preprocessing

In this section we present the different steps to extract, transform and integrate the source datasets into the main TERA components and sub-KGs. All data is transformed using custom mappings (scripts) from the sources to RDF triples. Table 5 shows an excerpt of the triples in TERA.

#### 5.2.1. Effects sub-KG construction

The effect data in ECOTOX consist of two parts, i.e., test definitions and results associated with the test definitions (see Tables 3 and 4, respectively). The important columns of a test are the chemical and the species used. Other columns include metadata, but these are optional and often empty. Each result is composed by
| # | Subject | Predicate | Object |
|---|---------|-----------|--------|
| 1 | et:test/1147366 | et:compound | et:chemical/134623 |
| 2 | et:test/1147366 | et:species | et:taxon/1 |
| 3 | et:test/1147366 | et:hasResult | et:result/102570 |
| 4 | et:result/102570 | et:endpoint | et:endpoint/LC50 |
| 5 | et:result/102570 | et:effect | et:effect/Mortality |
| 6 | et:taxon/1 | rdf:type | et:taxon/Pimephales |
| 7 | et:taxon/Pimephales | rdfs:subClassOf | et:taxon/Cyprinidae |
| 8 | et:taxon/1 | et:latinName | "Pimephales promelas" |
| 9 | et:taxon/1 | et:commonName | "Fathead Minnow" |
| 10 | et:taxon/1 | et:speciesGroup | et:group/Fish |
| 11 | et:taxon/1 | et:rank | et:rank/species |
| 12 | et:chemical/134623 | rdfs:label | "diethyltoluamide" |

| # | Subject | Predicate | Object |
|---|---------|-----------|--------|
| 13 | et:taxon/1 | owl:sameAs | ncbi:taxon/90988 |
| 14 | ncbi:taxon/90988 | owl:sameAs | wd:Q2700010 |
| 15 | wd:Q2700010 | owl:sameAs | eol:211492 |
| 16 | et:chemical/134623 | owl:sameAs | wd:Q408389 |
| 17 | wd:Q408389 | owl:sameAs | chembl_m:CHEMBL1453317 |
| 18 | wd:Q408389 | owl:sameAs | compound:CID4284 |
| 19 | wd:Q408389 | owl:sameAs | mesh:D003671 |
| 20 | wd:Q408389 | owl:sameAs | inchikey:MMOXZBCLCQITDF-UHFFFAOYSA-N |

| # | Subject | Predicate | Object |
|---|---------|-----------|--------|
| 21 | ncbi:taxon/90988 | rdf:type | ncbi:taxon/51137 |
| 22 | ncbi:taxon/90988 | rdf:type | ncbi:division/10 |
| 23 | ncbi:taxon/90988 | ncbi:scientific_name | "Pimephales promelas" |
| 24 | ncbi:taxon/90988 | ncbi:rank | ncbi:species |
| 25 | ncbi:division/51137 | rdfs:subClassOf | ncbi:taxon/7953 |
| 26 | ncbi:division/10 | rdfs:label | "Vertebrates" |
| 27 | ncbi:division/10 | owl:disjointWith | ncbi:division/1 |
| 28 | ncbi:division/10 | rdfs:label | "Invertebrates" |
| 29 | eol:211492 | eol:habitat | envo:0000153 |

| # | Subject | Predicate | Object |
|---|---------|-----------|--------|
| 30 | mesh:D003671 | mesh:broaderDescriptor | mesh:D001549 |
| 31 | mesh:D003671 | mesh:pharmacologicalAction | mesh:D007302 |
| 32 | chembl_m:CHEMBL1453317 | chembl:hasTarget | chembl_t:CHEMBL1907594 |
| 33 | chembl_t:CHEMBL1907594 | chembl:relSubsetOf | chembl_t:CHEMBL3137273 |
| 34 | compound:CID89845769 | vocab:hasParentCompound | compound:CID4284 |
| 35 | compound:CID131721069 | cheminf:CHEMINF_000478 | compound:CID4284 |
| 36 | compound:CID131721069 | rdf:type | bp:SmallMolecule |
| 37 | compound:CID7547 | vocab:is_active_ingredient_of | snomedct:411346009 |
| 38 | compound:CID131721069 | cheminf:CHEMINF_000480 | compound:CID10751691 |

Table 5

Example triples from the TERA knowledge graph. For space reasons, we have added the full id or label for some of the entities using footnote marks where 1inchikey:MMOXZBCLCQITDF-UHFFFAOYSA-N, 2Pimephales, 3Cyprinidae, 4Headwater, 5Benzamides, 6Insect Repellents, 7CHRNA3, 8CHRN4B, 9DETA-20, 10DETA Epichlorohydrin, 11Has component, 12Triclocarban, 13Trichlorocarbanilide-containing product, 14Similar to, 153-Chloromethyl-N,N-diethylbenzamide.
an endpoint, an effect, and a concentration (with a unit) at which the endpoint and effect are recorded.

This tabular data in ECOTOX is transformed into triples that form the effects sub-KG in TERA (KG_E). Note that a test can have multiple results. A subset of the effect triples are listed in Table 5 (see Triples $t_{11}$-$t_{12}$). A graphical representation for an effect test and its result is also shown in Figure 3.

ECOTOX contains metadata about the species and chemicals used in the experiments. This metadata is also included in TERA to facilitate the alignment with other resources (see Section 5.2.2).

(i) The ECOTOX metadata file species.txt includes common and Latin names, along with a (species) ECOTOX group (see triples $t_{8}$-$t_{10}$ in Table 5). This group is a categorization of the species based on ECOTOX use cases. Prefixes and abbreviations like sp., var. are removed from the label names.

(ii) The full hierarchical lineage is also available in the metadata file species.txt. Each column represents a taxonomic level, e.g., genus or family. If a column is empty, we construct an intermediate classification; for example, Daphnia magna has no genus classification in the data, then its classification is set to Daphniidae genus (family name + genus, actually called Daphnia). We construct these classifications to ensure the number of levels in the taxonomy is consistent (see triples $t_{6}$ and $t_{7}$ in Table 5). Note that when adding triples such as $t_{11}$ in Table 5, we also add a taxonomic rank to facilitate the querying for a specific taxonomic level.

(iii) The ECOTOX source file chemicals.txt includes chemical metadata and it is handled similarly to species.txt. The file includes chemical name (see $t_{12}$ in Table 5) and a (chemical) ECOTOX group.

For the units in the effect data, e.g., chemical concentrations (mg/L, mol/L, mg/kg, etc.), we reuse the QUDT 1.1 ontologies. When an unit such as mg/L is not defined, we define it according to Listing 1.

5.2.2. Alignment with state-of-the-art tools

ECOTOX database provides proprietary chemical identifiers (i.e., CAS numbers) and internal ECOTOX ids for species. In order to extrapolate effects across a larger set of chemicals and species than those available in ECOTOX, TERA integrates taxonomy and trait data from NCBI and EOL, and chemical data from PubChem, ChEMBL and MeSH.

Alignment between ECOTOX and the NCBI Taxonomy. There does not exist a complete and public alignment between the 23,439 ECOTOX species and the 1,830,312 NCBI Taxonomy species. We have used three methods, two state-of-art ontology alignments systems and a baseline, to align ECOTOX and the NCBI Taxonomy: (i) LogMap [33][44], (ii) AgreementMakerLight (AML) [25], and (iii) a string matching algorithm based on Levenshtein distance [45]. LogMap and AML were chosen since they have performed well across many datasets in the Ontology Alignment Evaluation Initiative (e.g., [2, 3, 61]). Most mappings in our setting are expected to be lexical, therefore, we also selected a purely lexical matcher to

---

19 As defined by U.S. EPA. Note that species hierarchies are contested among researchers.

20 QUDT 1.1: [http://linkedmodel.org/catalog/qudt/1.1/]

21 There are a total of 27,133 and 2,246,074 taxa in ECOTOX and NCBI, respectively. However, we focus on species, i.e., instances.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix qudt: <http://qudt.org/schema/qudt#> .
@prefix et: <https://cfpub.epa.gov/ecotox> .

```rdfs:label "Milligram per Liter"^^xsd:string ;
qudt:abbreviation "mg/L"^^xsd:string ;
qudt:conversionMultiplier 0.000001 ;
qudt:conversionOffset 0.0 ;
qudt:symbol "mg/dm^3"^^xsd:string .```

Listing 1: Unit definition of mg/L using QUDT.

| Method                  | 1-to-1 mappings        |
|-------------------------|------------------------|
|                         | # M        | R     | P≈  |
| LogMap                  | 20,585     | 0.81  | 0.87 |
| AML                     | 14,148     | 0.77  | 0.94 |
| String similarity (> 0.8) | 20,423     | 0.76  | 0.87 |
| Consensus (LogMap ∪ AML) | 12,740     | 0.76  | 0.98 |
| LogMap ∪ AML            | 21,145     | 0.83  | 0.86 |

Table 6
Alignment results for ECOTOX-NCBI. #M: number of mappings (at instance level), R: Recall, P≈: estimated precision.

evaluate if more sophisticated systems like LogMap and AML bring an additional value.

Due to the large size of the NCBI Taxonomy, we needed to split NCBI into manageable chunks to enable the use of ontology alignment systems. Fortunately, this can be easily done by considering the species division, e.g., mammal or invertebrate. This divides the NCBI Taxonomy into 11 distinct parts, which can be aligned to the taxonomy in ECOTOX.

Note that it is expected an entity from ECOTOX to match to a single entity in the NCBI Taxonomy, and vice-versa. Hence, 1-to-N and N-to-1 alignments were filtered according to the system computed confidence. A partial mapping curated by experts can be obtained through the ECOTOX Web. We have gathered a total of 2,321 mappings for validation purposes. Table 6 shows the alignment results over the ground truth samples for the 1-to-1 (filtered) system mappings. We report number of mappings (#M), Recall (R) and estimated precision (P≈) with respect to the known entities in the incomplete ground truth, assuming only 1-to-1 mappings are valid. P≈ is calculated as

\[ P≈ = \frac{|M_{ref} \cap M|}{|M_{ref}|} , \]

\[ M = \{ (e_e, owl:sameAs e_n) \in M \mid e_e \in \mathcal{E}_e \cap \mathcal{E}_n \land e_n \in \mathcal{E}_n \} , \]

where \(M_{ref}\) is the (incomplete) reference mapping set and \(M\) is the set of generated mappings between entities \(e_e \in \mathcal{E}_e\) from ECOTOX and entities \(e_n \in \mathcal{E}_n\) from the NCBI Taxonomy. \(\mathcal{E}_e \cap \mathcal{E}_n \subseteq \mathcal{E}_n \) and \(\mathcal{E}_e \cap \mathcal{E}_n \subseteq \mathcal{E}_e \) are the sets of entities that appear in the reference mappings. Thus, \(M_{ref}\) is defined as a subset of mappings from \(M\) involving entities in the reference mapping set \(M_{ref}\). Recall is defined in the standard way as

\[ R = \frac{|M \cap M_{ref}|}{|M_{ref}|} . \]

Note that, the recall will be the same for \(M\) and \(M_{ref}\).

We have selected the union of the 1-to-1 equivalence mappings computed by AML and LogMap to be integrated within TERA, as they represent the mapping set with the best recall with a reasonable estimated precision. This choice was made by considering the large uncertainty of downstream applications (effect prediction and risk assessment), where we prefer a larger coverage of the domain. See Triple [13] in Table 5 for an example of a system computed mapping between ECOTOX and the NCBI Taxonomy.

We use Wikidata as source of alignments between the NCBI Taxonomy and EOL, and among the used chemical datasets. Alignments are extracted via Wikidata’s query interface (i.e., SPARQL endpoint). The data in Wikidata concerning species and chemicals are

\[ ^{23}\text{ECOTOX interface: https://cfpub.epa.gov/ecotox/search.cfm} \]

\[ ^{24}\text{Wikidata endpoint: https://query.wikidata.org/sparql} \]
in large parts manually curated \cite{77} and will have a low error rate, comparatively to using the automated ontology alignment systems.

**Alignment between the NCBI Taxonomy and EOL.** In order to include in TERA trait data from EOL, we need to establish an alignment between EOL and the NCBI Taxonomy. We have constructed equivalence triples between the NCBI Taxonomy and EOL identifiers using Wikidata. The species identifiers are available as literals in Wikidata. Therefore, we concatenate them with the appropriate namespace. Listing 2 represents the SPARQL CONSTRUCT query used against the Wikidata endpoint. Here, we query Wikidata for instances of taxa, thereafter adding optional triple patterns for NCBI Taxonomy and EOL identifiers which are added as owl:sameAs triples to TERA.

Examples of resulting mapping triples are shown in \textit{t14} - \textit{t15} in Table 5. The proportion of species in Wikidata where this mapping exists is 49%.

**Alignment between chemical entities.** The mapping between ECOTOX chemical identifiers (CAS Registry Numbers) to Wikidata entities enables the alignment to a vast set of chemical datasets, e.g., PubChem, ChEBI, KEGG, ChemSpider, MeSH, UMLS, to name a few. The construction of equivalence triples between CAS, ChEMBL, MeSH, PubChem and Wikidata identifiers is shown in Listing 3. As for the case of species identifiers, the literal representing a chemical identifier is concatenated with the corresponding namespace. For the CAS Registry Numbers we also remove the hyphens to match ECOTOX notation. Examples of resulting mapping triples are shown in \textit{t16} - \textit{t20} in Table 5.

These mappings are not complete, but for some the coverage is large. Out of the chemicals used in ECOTOX, 73% have an equivalence in Wikidata (through the CAS registry numbers). Moreover, Wikidata chemicals has 4% ChEMBL identifiers, 0.5% MeSH identifiers, 55% PubChem identifiers, and 95% InChiKey identifiers.

### 5.2.3. Taxonomy sub-KG construction

The Taxonomy sub-KG (\textit{KG}S) integrates data from the NCBI Taxonomy and the EOL trait data. The integration of the NCBI Taxonomy into the TERA knowledge graph is split into several sub-tasks.

(i) We load the hierarchical structure included in the NCBI Taxonomy file \textit{nodes.dmp}. The columns of interest are the taxon identifiers of the child and parent taxon, along with the rank of the child taxon and the division where the taxon belongs. We use this to create triples like \textit{t21} - \textit{t22} and \textit{t24} - \textit{t25} in Table 5.

(ii) To aid alignment between the NCBI Taxonomy and the ECOTOX identifiers, we add the synonyms found in \textit{names.dmp}. Here, the taxon iden-
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>

CONSTRUCT { ?chemical owl:sameAs ?cas, ?chembl, ?mesh, ?pubchem, ?inchikey .}
WHERE {
?chemical wdt:P31 wd:Q11173 .
OPTIONAL {
  ?chemical wdt:P231 ?cas_id .
  BIND(iri(
    CONCAT("https://cfpub.epa.gov/ecotox/chemical/",
               REPLACE(?cas_id,'-',"")) AS ?cas)
  )
}
OPTIONAL {
  ?chemical wdt:P592 ?chembl_id .
  BIND(iri(
    CONCAT("http://rdf.ebi.ac.uk/resource/chembl/molecule/",
             ?chembl_id)) AS ?chembl)
  )
}
OPTIONAL {
  ?chemical wdt:P486 ?mesh_id .
  BIND(iri(
    CONCAT("http://id.nlm.nih.gov/mesh/",?mesh_id)) AS ?mesh)
  )
}
OPTIONAL {
  ?chemical wdt:P662 ?pubchem_id .
  BIND(iri(
    CONCAT("http://rdf.ncbi.nlm.nih.gov/pubchem/compound/CID",
             ?pubchem_id)) AS ?pubchem)
  )
}
OPTIONAL {
  ?chemical wdt:P235 ?inchikey_id .
  BIND(iri(
    CONCAT("https://rdf.ncbi.nlm.nih.gov/pubchem/inchikey/",
             ?inchikey_id)) AS ?inchikey)
  )
}
}

Listing 3: Construct chemical mapping between Wikidata and ECOTOX, ChEMBL, MeSH and PubChem. wdt:P31 is the predicate for instance of and wd:Q11173 is the class of all chemical compounds. wdt:P231, wdt:P592, wdt:P486, wdt:P662 and wdt:P235 are the relations for CAS Registry Number, ChEMBL ID, MeSH ID, PubChem CID and InChIKey, respectively.

tifier, its name and name type are used to create triples like \[t_{23}\] in Table 5. Note that a taxon in the NCBI Taxonomy can have several synonyms while a taxon in ECOTOX usually has two, i.e., common name and scientific name.

(iii) Finally, we add the labels of the divisions found in \[divisions.dmp\] (see triples \[t_{26}\] and \[t_{28}\]). We also add disjointness axioms among unrelated divisions, e.g., triple \[t_{27}\] in Table 5.

We use the TraitBank from EOL [58] to add species traits to TERA. The TraitBank is modeled as a property graph and can be accessed as a neo4j database or via a set of tabular files. To integrate the TraitBank into TERA we validate the identifiers used in EOL and convert to URIs. If an identifier is not a
valid URI, we replace invalid symbols. A trait example is shown as triple [f28] in Table 5. The EOL TraitBank also includes subsumption definitions (i.e., via rdfs:subClassOf) for a large portion of traits. These subsumptions can be downloaded separately and are added to TERA in a similar way as mentioned above.

5.2.4. Chemical sub-KG construction

The Chemical sub-KG (KG_C) is created from PubChem [38], ChEMBL [29], and MeSH [50]. These datasets are available for download as RDF triples. In addition, ChEMBL and MeSH can be accessed through the EBI and MeSH SPARQL endpoints, respectively.

The chemical subset of PubChem is used since information about chemicals is standardized in PubChem, while information about substances is not. In this subset we use: (i) component information, i.e., what are the building blocks of the chemical or parts of a mixture; (ii) type assertions, which either link to ChEBI or describe the type of molecule, e.g., small or large; (iii) role assertions, which describe additional attributes or relationships of the chemical, e.g., FDAApprovedDrug; and (iv) drug products, which link to the clinical data in SNOMED CT [7]. Examples of these can be seen in triples [f30], [f31], and [f32] in Table 5.

Parent chemical data in PubChem is limited to permutations e.g., bonds, polarity, and part of mixtures axioms (triple [f33]) in Table 5. Therefore, we use the hierarchical data about chemicals from MeSH. In addition to this data, we create similarity triples between chemicals. This is impractical to download, but can be calculated on demand. We add similarity triples to TERA where the Tanimoto (Jaccard) distance between the chemical fingerprints (gathered using PubChemPy [71]) is \( > 0.9 \) see triple [f34] in Table 5.

ChEMBL contains facts about bioactivity of chemicals. This contributes in assessing the danger of a chemical. In TERA, we use the mode of action (MoA) and target (receptor targeted by MoA; triple [f35]) in Table 5. These targets are organized in a hierarchy using chembli:relSubsetOf relations (see triple [f36]). The receptors will link to which organism it belongs to, however, we leave the inclusion of this information for future work.

We use the entire MeSH dataset in TERA. MeSH is organised as several hierarchies. The most prominent classifications are based on chemical groups and the intended use of the chemicals. Triples [f37] and [f38] in Table 5 show examples of chemical group and functional classifications.

5.3. TERA for data access

TERA covers knowledge and data relevant to the ecotoxicological domain and enables an integrated semantic access across data sets. In addition, the adoption of an RDF-based knowledge graph enables the use of an extensive range of Semantic Web infrastructure (e.g., reasoning engines, ontology alignment systems, SPARQL query engines).

The data integration efforts and the construction of TERA go in line with the vision in the computational risk assessment communities (e.g., Norwegian Institute for Water Research’s Computational Toxicology Program (NCTP)), where increasing the availability and accessibility of knowledge enables optimal decision making.

The knowledge in TERA can be accessed via predefined queries [29] (e.g., classification, sibling, and name queries, and fuzzy queries over the species names) and arbitrary SPARQL queries. The (final) output is flexible to the task, and can be given either as a graph or in tabular format. Listing 4 shows an example query to extract the chemicals and concentrations, at which, the species in the Oslofjord experience lethal effects.

5.4. TERA for effect prediction

TERA is used as background knowledge in combination with machine learning models for chemical effect prediction. TERA’s sub-KGs play different roles in effect prediction. The rich semantics of the species and chemical entities in the Taxonomy sub-KG (KG_T) and the Chemical sub-KG (KG_C), respectively, are embedded into low-dimensional vectors; while the Effects sub-KG (KG_E) provides the training samples for the prediction model. Each sample is composed of a chemical, a species, a chemical concentration, and the outcome or endpoint of the experiment. More details are given in Section 6, where the effect prediction model is built upon state-of-the-art knowledge graph embedding models.

Table 7 shows the sparsity-related measures of common benchmark datasets [27] and TERA’s KG_C and KG_T.

Grounded ingredients:
- Default value used in PubChem [27].
- Predefined queries are typically abstractions of SPARQL queries.
- YAGO3-10 [29], FB15k-237 [9], WN18 [17] and WN18RR [19].
the larger bias) in the triples in the KG [62], and indicate whether there are biases (the lower EE or RE, entity entropy (EE) and the relation entropy (RE) in-
and entities in the knowledge graph, respectively. The
where $RD = \frac{|T|}{|R|}$, and entity density, $ED = \frac{2|T|}{|E|}$, where $T$, $R$, and $E$ are the sets of triples, relations, and entities in the knowledge graph, respectively. The entity entropy (EE) and the relation entropy (RE) in-
dicate whether there are biases (the lower EE or RE, the larger bias) in the triples in the KG [62], and are
calculated as
\[
\begin{align*}
P(r) &= \frac{|t.p = r|}{|T|}, \\
P(e) &= \frac{|t.sb = e| + |t.ob = e|}{|T|}, \\
RE &= \sum_{r \in R} -P(r)\log(P(r)), \\
EE &= \sum_{e \in E} -P(e)\log(P(e)),
\end{align*}
\]
where $|t.p = r|$ is the number of triples with $r$ as predicate, and $|t.sb = e| + |t.ob = e|$ is the number triples with $e$ as subject or object.

In addition, we calculate the absolute density of the graph, which is $AD = \frac{|T|}{(|E|(|E| - 1))}$. This is the ratio of edges to the maximum number of edges possible in a simple directed graph [17].

High RD and low RE typically lead to a worse performance, while high ED and low EE often lead to better link prediction performance (e.g., [19]). In Table 7 we can see that the density and entropy values are in between those for YAGO3-10 and FB15k-237, which typically lead to worse and better predictive performance, respectively [19]. This shows that TERA is
6. Adverse biological effect prediction

The aim of chemical effect prediction is to extrapolate exiting data to new combinations of (possibly unknown) chemicals and species. In this section we present three classification models used to predict the biological effect of chemicals at the given concentration \( \kappa \). These models are fed with pre-trained knowledge graph embeddings (KGEs) and are appended to the baseline model fed with pre-trained KG embeddings. The output is a binary value that represents whether the chemical at the given concentration has a lethal effect on the species.

\[
y_c^0 = e_c, \quad y_s^0 = e_s, \quad y_{\kappa}^0 = \kappa
\]

Note that the effect can have a more fine-grained categorization (endpoints LCx, LDx, EC\(_x^{29}\) and NR-LETH in Table 3). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: “lethal” and “non-lethal”.

\[
y_{c,s,\kappa} = \begin{cases} 1 & \text{c is lethal to s at } \kappa, \\ 0 & \text{otherwise}. \end{cases}
\]

Note that the effect can have a more fine-grained categorization (endpoints LCx, LDx, EC\(_x^{29}\) and NR-LETH in Table 3). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: “lethal” and “non-lethal”.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

6.1. Baseline model

Our baseline prediction model is a multilayer perceptron (MLP) with multiple hidden layers. \( n_e \) hidden layers are appended to the embedding \( e_c \) of the chemical \( c \), \( n_s \) hidden layers are appended to the embedding \( e_s \) of species \( s \), and \( n_{\kappa} \) hidden layers appended to the real valued chemical concentration \( \kappa \). Thereafter, \( n \) hidden layers are further appended to the output of the previous hidden layers concatenated. Specifically, the model can be expressed by the following equations (with \( x_{c,s,\kappa} \) as input):

\[
y^0 = e_c, \quad y^0 = e_s, \quad y^0 = \kappa
\]

\[
y^0 = ReLu(y^{h-1}_c W^h_c + b^h_c), \quad h \in \{0, \ldots, n_e\}
\]

\[
y^0 = ReLu(y^{h-1}_s W^h_s + b^h_s), \quad h \in \{0, \ldots, n_s\}
\]

\[
y^0 = ReLu(y^{h-1}_{\kappa} W^h_{\kappa} + b^h_{\kappa}), \quad h \in \{0, \ldots, n_{\kappa}\}
\]

\[
y^0 = [y^c, y^s, y^\kappa]
\]

\[
y^h = ReLu(y^{h-1} W^h + b^h), \quad h \in \{1, \ldots, n\}
\]

\[
y^0 = \sigma(y^n W^n + b^n)
\]

\[
e_c, e_s, e_{\kappa} \in \mathbb{R}^k \text{ in } (9) \text{ denote the embeddings of } c \text{ and } s \text{ respectively, and are calculated as }
\]

\[
e_c = \delta_c W_c, \quad e_s = \delta_s W_s
\]

where \( \delta_c \) and \( \delta_s \) denote the one-hot encoding vectors of the chemical entity \( c \) (w.r.t. all the entities in \( \xi_c \) from \( KG_C \)) and the species entity \( s \) (w.r.t. all the entities in \( \xi_s \) from \( KG_S \)). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: “lethal” and “non-lethal”.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

\[c, s, \kappa\]

Notation. Throughout this section we use bold lower case letters to denote vectors while matrices are denoted as bold upper case letters. The vector representation of an entity and a relation are noted as \( e_c \) and \( e_r \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.
(i) **Simple setting.** Figure 4a shows the model without embedding transformation layers, i.e., \( n_c = n_s = n_e = 0 \) and \( n = 1 \).

(ii) **Complex setting.** The complex model shown in Figure 4b introduces transformation layers on the embeddings and chemical concentration input. These transformations aim at extracting the important information in the inputs and disregard the redundant information based on the output.

In the experiments we refer to the baseline models as **Simple one-hot** and **Complex one-hot**, depending on the selected MLP setting.

### 6.2. Baseline model with pre-trained KG embeddings

This models relies on pre-trained embeddings of chemicals and species computed using state-of-the-art KGE models (see Section 4.2 and Appendix A for an overview). A (different) KGE model is applied to the chemicals KG and the species KG.

These pre-trained KG embeddings are then given as input instead of the one-hot encoding vectors in the baseline model. We replace the trainable matrices \( \mathbf{W}_c \) and \( \mathbf{W}_s \) in Equation (16) by the matrices composed of embeddings by the respective KGE models. Namely \( \mathbf{W}_c \) is set to \( \left[ \mathbf{e}_{c,1}; \mathbf{e}_{c,2}; \ldots; \mathbf{e}_{c,|C|} \right] \), \( \mathbf{W}_s \) is set to \( \left[ \mathbf{e}_{s,1}; \mathbf{e}_{s,2}; \ldots; \mathbf{e}_{s,|S|} \right] \), where \( [::] \) denotes stacking vectors, \( \mathbf{e}_{c,j} \) denotes the embedding of the \( j \)th chemical in the chemicals \( KG_c \), \( \mathbf{e}_{s,i} \) denoted the embedding of the \( i \)th species in the species \( KG_s \).

In the experiments we refer to these models as **Simple PT KG\( E_C-KG\_S \)** and **Complex PT KG\( E_C-KG\_S \)**, depending on the selected MLP setting, where PT stands for pre-trained, and KG\( E_C \) and KG\( S \) are the KGE models used for the chemicals KG and the species KG, respectively (e.g., **Complex PT DistMult-HAKE**). For simplicity, we also refer to these models as PT-based models.

### 6.3. Fine-tuning optimization model

This model improves upon the pre-trained KG embeddings with fine-tuning based on the effect prediction data. This is done by simultaneously training the (selected) KGE models and the MLP-based baseline model. Such that the \( \mathbf{W}_c \) and \( \mathbf{W}_s \), and the MLP weights \( \mathbf{W}_\alpha \) and \( \mathbf{b} \), in Equations (10), (11), (14) and (15) are optimized simultaneously. Note that we initialize the KGE models with the previously pre-trained embeddings.

The model architecture is shown in Figure 5 and the overall loss to minimize is

\[
L = \alpha_C L_{KG_c} + \alpha_S L_{KG_s} + \alpha_{MLP} L_{MLP} \tag{17}
\]

where \( L_{KG_c} \) and \( L_{KG_s} \) respectively denote the loss of the chemical KG\( C \) and the species KG\( S \) when a specific KGE model is used\(^30\), \( \alpha_C \) and \( \alpha_S \) denote their weights respectively, \( L_{MLP} \) and \( \alpha_{MLP} \) denote the loss of the MLP and its weight. Specifically, we use binary cross-entropy (BCE) as the loss for the classification.

\[
L_{MLP} = \frac{1}{N} \sum_j y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j) \tag{18}
\]

\(^30\)Appendix A.5 introduces the used loss-functions in this work. The selection of the loss function for a KGE model will be via a hyper-parameter.
where $N$ denotes the size of the training samples, $y_i$ and $\hat{y}_i$ denote the sample label and the MLP output, respectively (as in Equation (8)). With the overall loss, gradient-based learning algorithms such as Adam optimizer [39] can be adopted to jointly training the embeddings of both KGEs and the MLP.

Figure 5 shows the full simultaneous fine-tuning model and the optimization process. The initial state of the entity lookups is the pre-trained embeddings. The full training procedure is summarised as follows:

1. Select $N$ triples from $KG_C$ and $KG_S$, where $N$ is the length of the effects training set.\(^{31}\)
2. Generate negative knowledge graph triples (see Appendix A.5 for details) from the extracted subsets of triples from $KG_C$ and $KG_S$, these negative $KG_S$ triples are referred to as $KG_C$ and $KG_S$.
3. Feed-forward the input through the model and calculate loss for each model component and combine according the loss weights.
4. Optimize the KG entity and relation embeddings, and the MLP layers.

These steps are repeated until the loss (only $L_{MLP}$) over the validation set stops improving.

In the experiments we refer to these models as Simple FT $KGE_C$-$KGE_S$ and Complex FT $KGE_C$-$KGE_S$, depending on the selected MLP setting, where FT stands for fine-tuning, and $KGE_C$ and $KGE_S$ are the KGE models used for the chemicals KG and the species KG, respectively (e.g., Simple FT HAKE-HAKE). For simplicity, we also refer to these models as FT-based models.
7. Results

7.1. Experimental setup

All models are implemented using Keras [16] and the model codes are available in our GitHub repository, alongside all data preparation and analysis scripts. [22]

7.1.1. Preparation of TERA for prediction

As shown earlier, TERA consists of three sub-KGs. These are the basis for the chemical effect prediction. [33] We process the sub-KGs further to limit their size by removing irrelevant triples for prediction. This is necessary to scale up the training of the KGE models. The reduction of TERA’s sub-KGs is performed according to the following steps:

(i) Effect data. For prediction purposes, the effect data in $KG_E$ is limited to four features, namely, chemical, species, chemical concentration, and effect. The chemical concentrations (κ, converted to mg/L) are log-normalized to remove the large discrepancy in scales. As mentioned, we separate the effects into two categories for simplicity, lethal and non-lethal effects. This reduces the possibility of ambiguity among the effects that do not cause death in the test species. We label lethal effects as 1 and non-lethal effects as 0.

(ii) $KG_C$. For each chemical in the effect data, we extract all triples connected to them using a directed crawl. This reduces the size of $KG_C$ to a manageable size for the KGE models. Moreover, we do not deem triples not directly connected to the effect data relevant for the prediction task, and may introduce unnecessary noise. As mentioned before, PubChem contains similarities between chemicals based on chemical fingerprints, however, for our use-case it is impractical to query them from the PubChem RDF data, therefore, we calculate similarity triples based on queried PubChem fingerprints. We use the same similarity threshold as PubChem, i.e., 0.9 [37].

(iii) $KG_S$. The same steps as for $KG_C$ are conducted for all species in the effect data.

A simple directed crawl over all predicates is sufficient to gather the interesting data in this setting as both $KG_C$ and $KG_S$ are primarily hierarchical and we start the crawls at the leaf nodes.

These steps reduce $KG_C$ to 241,442 triples and $KG_S$ to 59,673 triples. Some statistics of $KG_C$ and $KG_S$, and the reduced fragments $KG_C'$ and $KG_S'$, are given in Table 7 (Section 5.4). In the rest of the paper were refer to TERA’s reduced sub-KGs simply as $KG_C$ and $KG_S$.

The transformation from TERA’s $KG_C$ and $KG_S$ to model input is done by first dropping literals, thereafter assigning each entity an unique integer identifier which corresponds to the index of a column vector in matrices $W_c$ or $W_s$ in Equation (16), depending on which sub-KG is transformed. Relations are treated similarly.

7.1.2. Sampling

We use four sampling strategies of the effect data to analyze how the proposed classification models behave by varying the data parts that are used for training and testing. Note that, we only consider effect data where the chemical and species have mappings to external sources (e.g., NCBI Taxonomy and Wikidata, cf. Section 5.2.2) so that there is additional contextual information that can be used by the KGE models. For each of the strategies, the validation and test sets contain unseen chemical-organism pairs with respect to the training set. The strategies, however, differ with respect to the individual organism and chemical as follows:

Strategy (i) Random 70%/15%/15% training/validation/test split on the entire dataset (i.e., the chemicals and the organisms in the validation and test will most probably be known).

Strategy (ii) Training/validation/test split where there is no overlap between chemicals in the three sets (i.e., the chemicals in the validation and test sets are unknown). This resulted on a 77%/14%/9% split.

Strategy (iii) Training/validation/test split where there is no overlap between species in the three sets (i.e., the species in the validation and test sets are unknown). This resulted on a 77%/14%/9% split.

Strategy (iv) Training/validation/test split with no chemicals or species overlap in the three sets (i.e., both the chemicals and the organisms in the validation and test sets are unknown). This resulted on a 72%/14%/14% split.

\[ i \in [0, |E_C| - 1] \text{ for } KG_C \text{ and } i \in [0, |E_S| - 1] \text{ for } KG_S \]

https://github.com/NIWA-Knowledge-Graph/KGs_and_Effect_Prediction_2020

[33] All data used to create TERA was downloaded on the 14th of May 2020.
Note that since we use the species and chemicals as groups to divide the data rather than the samples, the splits can vary. For strategies (i-iii) there is a total of 14,377 effect data samples while for strategy (iv) the total number samples is 5,621. As above, this discrepancy is down to the way we split the data. We do not split across samples, but across chemicals and species. For example, some chemicals are used on (close to) all species, therefore, these chemicals are discarded in the sampling strategy (iv), affecting the final number of samples.

There were originally 57,560 samples, however, this includes experiment duplicates, i.e., same chemical, species, and endpoint, with different chemical concentrations. This is down to large discrepancies in laboratory testing variance, therefore, we use the median concentration across the duplicates. The prior probability is approximately 0.16/0.84 (i.e., ≈ 16% of samples are labelled as non-lethal and ≈ 84% of samples are labelled as lethal) across all sampling methods. We solve this when training by randomly oversampling the minority class until the prior probabilities are 0.5/0.5 in the training set. In this case, the oversampling is performed by adding duplicates samples labelled as non-lethal. Oversampling is a well-established technique used in many classification problems to remove bias during learning [11].

### 7.1.3. Hyper-parameters

To optimize the hyper-parameters for the KGE and classification models we use random search over the parameter ranges. We conduct 20 trials per model. Tables 8 and 9 contain the best hyper-parameters and can be used to reproduce the top performing models.

To find the best hyper-parameters for the KGE models, we use the loss as a proxy for performance, normalized by the initial loss, $R_{L_{ep}} = L_{ep}/L_0$, where $L_{ep}$ is the training loss at epoch $ep$, $L_0$ is the loss with the initial weights.

We use validation loss to select the best hyper-parameter setting for the classification models presented in Section 6.3. The best prediction models are re-fitted and evaluated 10 times to reduce the influence of initial conditions on the metrics. The average and standard deviation of the metrics are presented in Section 7.2.

The hyper-parameter ranges for the KGE models are shown in Table 8 based on common values used in the literature. We conduct 20 trials of random hyper-parameters choices and validate over the validation data. In Table 9 we show the best hyper-parameters.

We can see in Table 9 that the decomposition models have similar hyper-parameters for KGc and KGs. As shown in Section 5.4, the major difference between KGc and KGs is the relational density. Therefore, it is reasonable to believe that a lower relational density KG requires more parameters to have an equivalent representation in the embedding space. We can get the same observation for the geometric models except for TransE, where the embedding dimensions are similar. ConvE is more efficient in embedding dimension than ConvKB, however, since ConvE is slightly more complex than ConvKB this is expected. The difference in negative samples could be down to our implementation of ConvE, which varies from the original. Our implementation of all models relies on 1-to-1 scoring of triples, while the implementation of ConvE originally used 1-to-[|E|] scoring, where |E| is the number of entities in the KG [19].

The fine-tuning optimization model (Section 6.3) in order to save on intensive computation, reuses the same hyper-parameters found for the KGE models. Depending on the optimizer choice, the choice of loss weights, $\alpha_C$, $\alpha_S$, and $\alpha_{MLP}$, is important. However, our optimizer choice has dynamic learning rates per variable, and therefore, will adapt regardless of the loss weights and we can set $\alpha_C = \alpha_S = \alpha_{MLP} = 1$. Had we used, e.g., stochastic gradient descent, these variables would needed to be tuned.

### 7.1.4. Initialization of the fine-tuning optimization models

As presented in Section 6.3, we simultaneously train the KGE models and the MLP-based baseline model. This is done by initializing the model with (i) the weights learned in the correspondent baseline model with pre-trained embeddings, and (ii) the KG embed-
Loss function

\( L_{d_1} / L_{d_2} \)

6 / -

- / 5

6 / -

L

Sampling

14 / 20

- / 2

271 / 398

108 / 359

L

Embedding dimension

12 / 10

248 / 276

164 / 210

271 / 398

L

Bias

68 / 40

- / 6

163 / 372

27 / 42

- / 6

188 / 376

30 / 100

- / 6

226 / 196

23 / 57

14 / 20

108 / 130

56 / 13

14 / 20

248 / 276

18 / 90

- / 6

228 / 196

68 / 40

- / 6

143 / 383

28 / 43

- / 6

163 / 372

27 / 42

- / 6

188 / 376

30 / 100

- / 6

226 / 196

23 / 57

14 / 20

108 / 130

56 / 13

14 / 20

248 / 276

18 / 90

- / 6

228 / 196

68 / 40

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{Model} & \textbf{Loss function} & \textbf{Margin} & \textbf{Bias} & \textbf{Embedding dimension} & \textbf{Negative Samples} \\
\hline
DistMult & \( L_{d_1} / L_{d_2} \) & - / 2 & - & 143 / 383 & 28 / 43 \\
Complex & \( L_{d_1} / L_{d_2} \) & - / 4 & - & 163 / 372 & 27 / 42 \\
HolE & \( L_{d_1} / L_{d_2} \) & 6 / - & - & 188 / 376 & 30 / 100 \\
TransE & \( L_{d_1} / L_{d_2} \) & 4 / 7 & 14 / 20 & 226 / 196 & 23 / 57 \\
RotatE & \( L_{d_1} / L_{d_2} \) & 5 / 2 & 16 / 6 & 271 / 398 & 75 / 22 \\
pRotatE & \( L_{d_1} / L_{d_2} \) & - / - & 14 / 16 & 164 / 210 & 34 / 82 \\
AKE & \( L_{d_1} / L_{d_2} \) & - / - & 12 / 10 & 108 / 359 & 56 / 13 \\
ConvKB & \( L_{d_1} / L_{d_2} \) & - / 5 & - & 248 / 276 & 18 / 90 \\
ConvE & \( L_{d_1} / L_{d_2} \) & 7 / 3 & - & 228 / 196 & 68 / 40 \\
\hline
\end{tabular}
\end{center}
\caption{Best hyper-parameters for KGE models. The two values before and after \( / \) are for the embeddings of \( KG_C \) and \( KG_S \), respectively.}
\end{table}

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{Model} & \textbf{Sampling} & \textbf{\# units} \\
\hline
Complex one-hot & (i) & (128)/(128)/−/− \\
& (ii) & (128)/(256)/(8,8)/− \\
& (iii) & (256,128)/(128)/(4,4,4)/− \\
& (iv) & (256,256)/(128)/(8,8)/(128) \\
Complex PT DistMult-HAKE (top-1 in (ii)) & (i) & (256,256)/(256)/(16,4)/(512,64) \\
Complex PT HolE-ConvKB (top-1 in (ii)) & (ii) & (512,128,128)/(512)/−/−/(64) \\
Complex PT HAKE-DistMult (top-1 in (iii,iv)) & (iii) & (64)/(512)/(16,32)/(16) \\
& (iv) & (128)/−/(4,8,8)/(256,128) \\
\hline
\end{tabular}
\end{center}
\caption{Number of units in the hidden layers in the (complex) one-hot model and the top-1 prediction models with pre-trained KG embeddings. The same parameters are used for the fine-tuning models. Organized as follows: \((b_1,\ldots,b_n)\)/\((b_1',\ldots,b_n')\)/\((b_1'',\ldots,b_n'')\)/\((b_1'',\ldots,b_n'')\)/\((b_1'',\ldots,b_n'')\) as in Equations \((6,7,8,12)\) and \((14)\). \(−\) denotes no hidden layers. e.g., \((128)/(256)/(8,8)/−\) denotes \(n_C=1, n_S=1, n_K=2, n_0=0\) and \(b_1=128, b_2=256, b_3=8\) and \(b_4=8\).}
\end{table}

dings learned with the respective KGE models. For example, the \textit{Complex PT DistMult-HAKE} model is initialized with the learned weights with the \textit{Complex PT DistMult-HAKE} model and the pre-trained KG embeddings using DistMult and HAKE models. Then the model is further trained with a small learning rate. We found that reducing the learning rate by a factor of 100 worked well. Using this learning rate we optimize the model until convergence.

7.1.5. Simple and complex settings

As presented in Section \([6,7]\), we use two settings in our classification models: simple and complex. This will help us isolate the effects of the KG embeddings versus the power of the MLP model. The simple setting uses no branching layers, i.e., \(n_C=n_S=n_K=0\) and \(n=1\) as in Equations \((10,11,12)\) and \((14)\) with 128 units in the hidden dense layer. For the complex models we use random search (20 trials) to find the optimal number of layers and units out of the ranges shown in Table \([8]\). The optimal choices for the top-performing models (using one-hot and pre-trained embeddings) are shown in Table \([10]\).

Looking at the increasing complexity of the layer configuration of the one-hot models in Table \([10]\), we can see a correlation from the simplest sampling strategy \((i.e., (ii))\) through the most challenging one \((i.e., (iv))\). The same can be seen for PT HAKE-DistMult from strategy \((iii)\) to \((iv)\), where the number of layers increase. Overall we can see that the layer configurations of the chemical branch is more complex than for the species branch. This indicates that the KGE models are better at representing \(KG_S\) than \(KG_C\).

7.2. Prediction results

In this section we present a summary of the conducted chemical effect prediction evaluation. Complete results are available at the project repository.\footnote{https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020}
The default decision threshold is set to 0.5. That is, if a model predicts \( \hat{y} > 0.5 \) for an input \( x_{c,s,k} \) then the chemical \( c \) is considered lethal to \( s \) at a concentration \( k_5 \). We use several metrics to compare the different prediction models. These are Sensitivity (i.e., recall), Specificity, and Youden’s index \( (YI) \). Precision and F-score were also considered as metrics. However, they were not representative for the performance with respect to non-harmful chemicals. This is attributed to the larger number of positive samples (i.e., harmful chemicals) than negative samples (i.e., non-harmful chemicals) in the test data.

Sensitivity and Specificity are defined as

\[
\text{Sensitivity} = \frac{TP}{TP + FN}, \quad (19)
\]

\[
\text{Specificity} = \frac{TN}{FP + TN}, \quad (20)
\]

where TP, FN, TN, and FP are true positives, false negatives, true negatives and false positives, respectively. \( YI \) is defined as

\[
YI = \text{Sensitivity} + \text{Specificity} - 1. \quad (21)
\]

We also present the maximized Youden’s index \( (YI_{\text{max}}) \), this is defined as

\[
YI_{\text{max}} = \max_{\tau} \text{Sensitivity} + \text{Specificity} - 1, \quad (22)
\]

\( i.e., \) we maximize Youden’s index based on the decision threshold \( (\tau) \), we call this optimal threshold \( \tau_{\text{max}} \). This metric is equivalent to the maximum of the Receiver operating characteristic (ROC) curve over a random model and can be used to select the optimal decision threshold in a production environment (based on validation data). We do not present ROC (or area under ROC, AUC) as a metric as it correlates \( (> 0.99) \) with \( YI_{\text{max}} \) in our case.

In our setting, sensitivity is a measure on how well the models identify harmful chemicals while specificity measures models’ ability to identify non-harmful chemicals. Youden’s index is used to capture the usefulness of a diagnostic test (or in our case, a toxicity test). A useless test will have \( YI = 0 \) while with \( YI > 0 \) a test is useful. \( YI \) is also thought of as how well informed a decision might be. Note that, \( YI \) can be less than 0, but this is solved by swapping labeled classes. Similarly to how negative correlation is still useful.

Tables 11-14 show the results for each of the data sampling strategies (i)-(iv), respectively. The tables include the three best models (based on \( YI \)) for the baseline model using one-hot and pre-trained (PT) KG embeddings, and the fine-tuning (FT) models using the same combination of KGE models as the selected PT-based models. We have also included a model with middling performance (i.e., 40 out of 81 models) and the worst performing model. Note that for the PT- and FT-based models we have evaluated 81 combinations KGE\(_C\)–KGE\(_S\) of KGE models. All models were evaluated using the simple and complex MLP settings. For example, the model Complex FT DistMult-HoLE denotes that fine-tuning was used together with the complex MLP setting, and DistMult was selected to embed the chemicals KG\(_C\) while HolE was used to embed the species KG\(_S\). We present the mean and standard deviation over 10 evaluation runs, \( i.e., \) we re-initialize and re-train the models 10 times. Results highlighted in bold are the best mean results of the corresponding metrics. Underlined results are where there is a \( > 32\% \) chance that a single run outperforms the best mean \( (i.e., \) one standard deviation contains 68% of results, assuming normally distribute results).

Overall, models with the complex setting and fine-tuning are needed as the data sampling strategies become more challenging. Moreover, all models favour sensitivity over specificity at default decision threshold \( (0.5) \). This is down to the imbalance in the data. We can see the imbalance by \( \tau_{\text{max}} \), it is \( > 0.5 \) for most models. As we use a log-loss instead of a discrete loss, this is to be expected for imbalanced data.

For settings (iii) and (iv) the performance drops and the standard deviation increases compared to the other strategies. This large standard deviation leads to large overlaps in quantiles among top-3 models in all categories, such that, by chance, one of these models could perform best in one individual evaluation.

7.2.1. One-hot baseline models

For the sampling strategy (i) the one-hot baseline models perform well, especially, with the complex one-hot model. This complex model is equivalent in terms of \( YI \) as the best simple pre-trained model. The

---

\[ ^{36} \] We set the decision threshold \( \hat{y} > 0.5 \) since the model output bias \( (c.f. \text{Equation} (15)) \) will be (close to) 0.5 after training. Recall that we have oversampled the classes to reach a 0.5/0.5 prior probability during training \( (c.f. \text{Section 1.7.2}) \).

\[ ^{37} \] Note that we only consider the best mean result and not the standard deviation in both directions.
story is largely the same in setting (ii), where the complex one-hot model performs within 1.5% of the best simple pre-trained models. With strategies (iii) and (iv) the one-hot models degrade, especially in strategy (iv) where the Youden’s index is near zero (< 0.1). This is expected as the one-hot baseline models lack important background information about the entities, specially for unseen chemicals and species, that the KG embedding models aim at capturing.

7.2.2. Baseline with pre-trained KG embeddings

We can see that the PT-based models do not lead to an important improvement with respect to \(Y_{I_{\text{max}}}\) in sampling strategy (i). The top-1 complex PT model, however, yields a better balance between sensitivity and specificity leading to an improved \(Y_I\) over the complex one-hot models. The two middling performing models, Simple \(PT\) \(p\text{-}\text{RotateE-ConvE}\) and Complex \(PT\) \(\text{ComplEx-DistMult}\), still retain a decent level of performance.

The results with the strategy (ii) are similar to strategy (i), the delta in \(Y_I\) between the simple and the complex PT-based models are about 5%. This slight improvement is due to the increased balance between sensitivity and specificity which in turn leads to a higher \(Y_I\).

In the sampling strategy (iii) we can observe that the improvement of the PT-based models over the one-hot models increases. The increase is up to 25% in \(Y_I\) of the the best PT-based model over the best one-hot model. In addition, we observe in this strategy that the standard deviation increases, especially in specificity, leading to a large portion of the models that are within one standard deviation of the best model in terms of \(Y_I\).

Finally, the impact of using a PT-based models is strengthen in strategy (iv). The delta between the one-hot and PT-based models is up to 40% in \(Y_I\), and larger for \(Y_{I_{\text{max}}}\). We see that all models struggle with specificity in this setting, this is down to the difficulty of predicting true negatives. This also leads to a larger variation, with certain models yielding standard devia-
in the same order of magnitude as the metric (e.g., Simple FT HAKE-ComplEx).

7.2.3. Fine-tuning optimization model

The FT-based models, with some exceptions, improve the results over the PT-based models, most notably in sampling strategies (iii) and (iv). For example, the FT-based models Complex FT HolE-DistMult and Simple FT HolE-ComplEx are the best models in terms of $YI$ and $YI_{\text{max}}$ in strategy (iv), respectively. We can also see in strategies (i) and (ii) that the FT-based models improve middling and worst performing PT-based models, e.g., Simple FT RotatE-ConvE in strategy (i) improves from $YI = 0.21$ to $YI = 0.726$ using fine-tuning of the KG embeddings. The results are expected as the fine-tuned KG embeddings are tailored to the effect prediction task.

7.3. KG embedding analysis

In this section we look at correlations between KG model choices and prediction performance. KGE models are designed to capture certain structures in the data, and this can give some explanation of which parts of the KGs are important for prediction.

First, in Table 12 we show how many times a KGE model is used when regarding the top 10 performing combinations (out of the total 81 possible combinations). We focus on the choices when using the simple MLP setting to reduce the influence of the non-linear transforms on the embeddings.

Looking at Table 12 we can see that the KGE models used to embed the chemicals $KG_C$ in the best performing models is distributed evenly across most models and settings. This indicates that the performance of the prediction models is not highly correlated with the use of a KGE model on $KG_C$. Referencing Table 7, the high relational density in $KG_C$ can contribute to worse performance [62] and therefore equal distribution of models in Table 12. This is different for $KG_S$. For sampling strategies (i) and (ii), HAKE is extensively used in the top models to embed $KG_S$. HAKE is designed to embed hierarchies. Therefore, this indicates that in strategies (i) and (ii) the hierarchical structure of $KG_S$ dwarfs the rest of the KG. $KG_S$ has a higher entity
density and lower entity entropy (Table 7) than $KG_C$. This should lead to higher performance generally, but might also lead to larger discrepancies between models as seen in Table 15.

The use of the decomposition models increase in strategies (iii) and (iv) for the embedding of $KG_S$, which indicates that KG structures, other than the hierarchy, are important. Overall, DistMult and ComplEx can be used to great effect in strategies (iii) and (iv) while the geometric model, HAKE, is more successful in the less challenging strategies (i) and (ii).

7.3.1. Explained variance

Explained variance is a measure of how many principal components are required to describe all components. In Figure 6, we present how the $YI$ metric depends on the explained variance of the top-10 principal components (i.e., $\sum_{i=1}^{10} p\{CA\}_i$). We show all (81 per sampling strategy) PT-based prediction model results, simple MLP setting in Figure 6a and complex setting in Figure 6b. For example, in Figure 6a, the best model in the strategy (iv), Simple PT pRotatE-ComplEx have explained variance of 0.49 compared to the worst model, Simple PT HAKE-HAKE, with explained variance of 0.34. Coincidently, these two points does not follow the trend lines in these figures which indicate negative correlation between $YI$ and explained variance. The trend lines can be interpreted in two ways. First, it is counter-intuitive as we would expect more descriptive embeddings, i.e., larger explained variance, to perform better. On the other hand, the top-10 principal components may not be representative enough to capture the semantics of the KG embeddings, and thus, a large explained variance does not necessarily correlate with a high performance.

Figure 7 represents the explained variance against sensitivity. We can see that the trend is flat for strategy (iv), but positive for strategies (i)-(iii). This means that the trends in Figure 6 are explained by specificity rather than sensitivity. By balancing sensitivity and specificity, i.e., $YI_{max}$ as seen in Figure 8, the rate of change is reduced compared to $YI$ in Figure 6.

| Model                        | Sensitivity | Specificity | $YI$   | $YI_{max}$ | $\tau_{max}$ |
|------------------------------|-------------|-------------|--------|------------|--------------|
| Simple one-hot               | 0.822 ± 0.058 | 0.439 ± 0.054 | 0.261 ± 0.058 | 0.31 ± 0.047 | 0.597 ± 0.182 |
| Simple PT ConvKB-DistMult    | 0.966 ± 0.007 | 0.626 ± 0.047 | 0.591 ± 0.045 | 0.623 ± 0.049 | 0.67 ± 0.058  |
| Simple PT HAKE-DistMult      | 0.958 ± 0.023 | 0.628 ± 0.026 | 0.586 ± 0.033 | 0.626 ± 0.045 | 0.613 ± 0.092 |
| Simple PT ConvKB-TransE      | 0.969 ± 0.009 | 0.614 ± 0.048 | 0.583 ± 0.024 | 0.642 ± 0.011 | 0.643 ± 0.059 |
| Simple PT ConvE-RotatE       | 0.934 ± 0.055 | 0.276 ± 0.026 | 0.209 ± 0.043 | 0.273 ± 0.071 | 0.596 ± 0.13  |
| Simple PT HolE-HAKE          | 0.88 ± 0.089  | 0.115 ± 0.083 | −0.005 ± 0.075 | 0.077 ± 0.057 | 0.783 ± 0.18  |
| Simple PT ConvKB-DistMult    | 0.947 ± 0.014 | 0.667 ± 0.02  | 0.614 ± 0.013 | 0.645 ± 0.011 | 0.736 ± 0.087 |
| Simple PT HAKE-DistMult      | 0.947 ± 0.012 | 0.662 ± 0.035 | 0.609 ± 0.031 | 0.634 ± 0.026 | 0.701 ± 0.132 |
| Simple PT ConvKB-TransE      | 0.934 ± 0.009 | 0.68 ± 0.018  | 0.615 ± 0.014 | 0.642 ± 0.015 | 0.687 ± 0.065 |
| Simple PT ConvE-RotatE       | 0.915 ± 0.013 | 0.454 ± 0.028 | 0.369 ± 0.027 | 0.402 ± 0.028 | 0.658 ± 0.083 |
| Simple PT HolE-HAKE          | 0.931 ± 0.009 | 0.118 ± 0.036 | 0.049 ± 0.038 | 0.171 ± 0.038 | 0.882 ± 0.127 |
| Complex one-hot              | 0.796 ± 0.028 | 0.571 ± 0.041 | 0.367 ± 0.054 | 0.398 ± 0.043 | 0.526 ± 0.076 |
| Complex PT HAKE-DistMult     | 0.969 ± 0.016 | 0.642 ± 0.044 | 0.61 ± 0.034  | 0.643 ± 0.026 | 0.675 ± 0.105 |
| Complex PT pRotatE-ComplEx   | 0.929 ± 0.024 | 0.668 ± 0.048 | 0.597 ± 0.048 | 0.62 ± 0.046  | 0.526 ± 0.145 |
| Complex PT ConvKB-DistMult   | 0.965 ± 0.013 | 0.631 ± 0.078 | 0.597 ± 0.07  | 0.62 ± 0.039  | 0.597 ± 0.149 |
| Complex PT ComplEx-HolE      | 0.991 ± 0.01  | 0.237 ± 0.106 | 0.228 ± 0.098 | 0.45 ± 0.028  | 0.721 ± 0.047 |
| Complex PT ComplEx-HAKE      | 0.9 ± 0.055  | 0.097 ± 0.047 | −0.003 ± 0.064 | 0.133 ± 0.081 | 0.696 ± 0.22  |
| Complex FT HAKE-DistMult     | 0.932 ± 0.011 | 0.69 ± 0.024  | 0.622 ± 0.023 | 0.652 ± 0.022 | 0.706 ± 0.134 |
| Complex FT pRotatE-ComplEx   | 0.931 ± 0.025 | 0.672 ± 0.042 | 0.602 ± 0.045 | 0.631 ± 0.037 | 0.627 ± 0.157 |
| Complex FT ConvKB-DistMult   | 0.953 ± 0.008 | 0.642 ± 0.027 | 0.596 ± 0.027 | 0.625 ± 0.028 | 0.753 ± 0.138 |
| Complex FT ComplEx-HolE      | 0.898 ± 0.035 | 0.591 ± 0.064 | 0.489 ± 0.042 | 0.521 ± 0.027 | 0.612 ± 0.156 |
| Complex FT ComplEx-HAKE      | 0.88 ± 0.032  | 0.255 ± 0.026 | 0.135 ± 0.034 | 0.204 ± 0.06  | 0.775 ± 0.268 |

Table 13
Prediction results for sampling strategy (iii). Same notation as Table 11.

---

38 We use the scikit-learn implementation [60] based on [72].
| Model                      | Sensitivity | Specificity | YI    | YI_{max} | t_{max} |
|---------------------------|-------------|-------------|-------|----------|---------|
| Simple one-hot            | 0.612 ± 0.096 | 0.421 ± 0.107 | 0.033 ± 0.14 | 0.113 ± 0.076 | 0.555 ± 0.306 |
| Simple PT HAKE-ComplEx    | 0.971 ± 0.011 | 0.361 ± 0.065 | 0.332 ± 0.056 | 0.546 ± 0.031 | 0.89 ± 0.042 |
| Simple PT pRotatE-ComplEx | 0.972 ± 0.008 | 0.36 ± 0.079  | 0.332 ± 0.074 | 0.527 ± 0.045 | 0.852 ± 0.04 |
| Simple PT HolE-ComplEx    | 0.965 ± 0.032 | 0.363 ± 0.068 | 0.328 ± 0.063 | 0.549 ± 0.075 | 0.856 ± 0.077 |
| Simple PT pRotatE-RotatE | 0.917 ± 0.011 | 0.168 ± 0.016 | 0.084 ± 0.013 | 0.151 ± 0.021 | 0.779 ± 0.182 |
| Simple PT HAKE-HAKE       | 0.8 ± 0.095   | 0.128 ± 0.066 | −0.072 ± 0.07 | 0.033 ± 0.027 | 0.736 ± 0.321 |
| Simple FT HAKE-ComplEx    | 0.963 ± 0.011 | 0.423 ± 0.102 | 0.386 ± 0.096 | 0.57 ± 0.03  | 0.875 ± 0.079 |
| Simple FT pRotatE-ComplEx | 0.954 ± 0.009 | 0.5 ± 0.058  | 0.454 ± 0.052 | 0.569 ± 0.024 | 0.854 ± 0.073 |
| Simple FT HolE-ComplEx    | 0.965 ± 0.007 | 0.418 ± 0.058 | 0.383 ± 0.053 | 0.571 ± 0.042 | 0.9 ± 0.046  |
| Simple FT pRotatE-RotatE | 0.806 ± 0.039 | 0.229 ± 0.027 | 0.035 ± 0.016 | 0.131 ± 0.032 | 0.782 ± 0.157 |
| Simple FT HAKE-HAKE       | 0.893 ± 0.046 | 0.104 ± 0.051 | −0.003 ± 0.031 | 0.037 ± 0.033 | 0.588 ± 0.332 |
| Complex one-hot           | 0.656 ± 0.069 | 0.422 ± 0.075 | 0.078 ± 0.053 | 0.124 ± 0.036 | 0.645 ± 0.178 |
| Complex PT HAKE-DistMult  | 0.923 ± 0.013 | 0.434 ± 0.059 | 0.357 ± 0.052 | 0.488 ± 0.074 | 0.808 ± 0.07 |
| Complex PT HolE-DistMult  | 0.949 ± 0.016 | 0.38 ± 0.084  | 0.33 ± 0.076  | 0.443 ± 0.089 | 0.805 ± 0.07 | 0.817 ± 0.052 |
| Complex PT ConvKB-DistMult| 0.942 ± 0.01  | 0.387 ± 0.038 | 0.329 ± 0.039 | 0.484 ± 0.066 | 0.817 ± 0.052 |
| Complex PT HolE-RotatE    | 0.932 ± 0.014 | 0.15 ± 0.018  | 0.082 ± 0.023 | 0.168 ± 0.015 | 0.861 ± 0.064 |
| Complex PT TransE-HAKE    | 0.756 ± 0.047 | 0.19 ± 0.077  | −0.054 ± 0.089 | 0.057 ± 0.046 | 0.742 ± 0.253 |
| Complex FT HAKE-DistMult  | 0.925 ± 0.021 | 0.513 ± 0.064 | 0.437 ± 0.058 | 0.522 ± 0.034 | 0.83 ± 0.09 |
| Complex FT HolE-DistMult  | 0.926 ± 0.015 | 0.536 ± 0.03  | 0.462 ± 0.03  | 0.543 ± 0.039 | 0.81 ± 0.084 |
| Complex FT ConvKB-DistMult| 0.933 ± 0.01  | 0.525 ± 0.065 | 0.459 ± 0.063 | 0.55 ± 0.04  | 0.746 ± 0.122 |
| Complex FT HolE-RotatE    | 0.863 ± 0.057 | 0.194 ± 0.053 | 0.057 ± 0.015 | 0.11 ± 0.021 | 0.81 ± 0.278 |
| Complex FT TransE-HAKE    | 0.892 ± 0.027 | 0.075 ± 0.043 | −0.033 ± 0.049 | 0.072 ± 0.048 | 0.958 ± 0.077 |

Table 14
Prediction results sampling strategy (iv). Same notation as Table 11.

| KGE model | # uses (i) | # uses (ii) | # uses (iii) | # uses (iv) |
|-----------|------------|-------------|--------------|-------------|
| DistMult  | 1/0        | 0/1         | 1/7          | 0/4         |
| ComplEx   | 1/1        | 1/3         | 2/1          | 1/5         |
| HolE      | 2/0        | 1/0         | 1/0          | 1/0         |
| Total decomposition | 4/1 | 2/4 | 4/8 | 2/9 |
| TransE    | 1/0        | 2/0         | 1/2          | 0/0         |
| RotatE    | 0/0        | 0/0         | 0/0          | 1/0         |
| pRotatE   | 1/0        | 1/0         | 1/0          | 3/0         |
| HAKE      | 2/8        | 3/5         | 1/0          | 2/0         |
| Total geometric | 4/8 | 6/5 | 3/2 | 5/0 |
| ConvKB    | 1/1        | 0/1         | 2/0          | 0/1         |
| ConvE     | 1/0        | 2/0         | 1/0          | 2/0         |
| Total convolutional | 2/1 | 2/1 | 3/0 | 2/1 |

Table 15
Usage of KGE models for each sampling strategy in simple MLP setting in top-10 performing combinations. Note that, there is one model for the KG_{C} and one for KG_{S}, such that there is a total of 20 models per sampling strategy. Notation: ‘used in KG_{C}’ ‘used in KG_{S}, e.g., HAKE, 2/8 in sampling strategy (i), indicates that HAKE is used to embed KG_{C} 2 out of top-10 combinations and it is used to embed KG_{S} 8 out of top-10 combinations.
7.4. Example predictions

Table 16 shows a few examples of correct (TP and TN) and incorrect predictions (FN and FP).

Benthiocarb and permethrin are both biocides with different targets: benthiocarb is a herbicide and permethrin is an insecticide. It is therefore not surprising that benthiocarb has a low predicted effect on sea urchins, while permethrin has a severe effect on bivalves.

There are several possible explanations for the failed predictions. A wrong prediction of potassium chloride toxicity to a marine copepod (Megacyclops viridis)
could be due to the prediction model not being accurate enough for metal salts, or the copepod species being particularly sensitive to changes in osmolarity due to salt content. The wrong prediction of lack of herbicide toxicity \( (i.e., \text{carfentrazone-ethyl}) \) to a flower \( (i.e., \text{eudicots}) \) could be due to the fact that flowers, and plants in general, are severely underrepresented in the available effect prediction data.

8. Discussion

We have introduced the Toxicological Effect and Risk Assessment (TERA) knowledge graph and shown how we can directly use it in chemical effect prediction. The use of TERA improves the PT-based prediction models over the one-hot baselines. In the most challenging data sampling strategies, we have also seen the benefits of creating tailored \( (i.e., \text{fine-tuned}) \) KG embeddings in the FT-based prediction models.

8.1. TERA knowledge graph

The constructed knowledge graph consists of several sources from the ecotoxicological domain. There are three major parts in TERA: the effects data, the chemical data, and the species taxonomic data. Integrating each part has different challenges. The chemical and pharmacological communities have come a long way in annotating their data as knowledge graphs and ontologies. Here, selecting the correct subsets to work with the chemical effect prediction data was a major challenge. This had to be done based on mappings between effect data and chemical data that were extracted from Wikidata. We selected a relatively small subset of the chemical sub-KG to facilitate faster
model training, however, still larger than the extracted fragment from the species sub-KG. The species sub-KG was created from tabular data and cleaned by removing several annotation labels with redundant information. This sub-KG was aligned using ontology alignment systems to the species taxonomy in the effects sub-KG. This required pre-processing of the KG, where it was divided into smaller parts such that the selected systems could perform the alignment. We used several standard ontologies to facilitate the transformation of the effect data into a knowledge graph. This involved not only automatic processes, but also an important amount of manual work.

Integrating more data into TERA involves the creation of mappings to the existing data. This is possible for a large amount of chemical datasets as Wikidata links multiple datasets, e.g., the chemical compound diethyltoluamide (wd:Q408389) has ~35 distinct identifiers. Biological data, both taxonomic and effects, might be harder to align to TERA as these mappings are not available in Wikidata. Here, ontology alignment systems play an important role to fill this gap.

The additional integrated data will give larger coverage of the domain, and thereby, improve model performance. However, adding more data will also increase the memory and time requirements of KGE models. This was bypassed in this work by reducing TERA to only relevant parts.

Adding additional domain knowledge is also critical in other applications, such as using TERA for data access.

8.2. Performance of prediction models

We have shown that the ability to embed some structure types of different KGE models largely impact the prediction models. We see that some KGE models fail to capture the semantics of the chemicals and the species, which leads to similar performance to the one-hot baselines. Moreover, in a few isolated cases the performance is reduced further which leads us to believe that the embeddings collapse in one or some dimensions, making it impossible to distinguish among entities.

We suspect that the even distribution of KGE models to embed $KG_C$ (Table 15) in most settings is likely down to the structure of $KG_C$. This sub-KG has, unlike $KG_S$’s tree structure, a forest structure, and models that can deal with trees (as in $KG_S$) fail here, e.g., an entity in $KG_C$ can have multiple parents, but only one grand-parent. In this case, some models may create very similar or the same embeddings for the parent nodes.

9. Conclusions and future work

TERA is a novel knowledge graph which includes large amounts of data required by ecological risk assessment. We have conducted an extensive evaluation of KGE embedding models in a novel and very challenging application domain. Moreover, we have shown the value of using TERA in an ecotoxicological effect prediction task. The fine-tuning optimization model architecture to adapt the KG embeddings to the prediction task has, to our knowledge, not been applied elsewhere.

9.1. Value for the ecotoxicology community

The creation of TERA is of great importance to future effect modelling and computational risk assessment approaches within ecotoxicology. Where the strategic goal is designing and developing prediction models to assess the hazard and risks of chemicals and their mixtures where traditional laboratory data cannot easily be acquired.

A great effort in the hazard and risk assessment of chemicals is the reduction of regulatory-mandated animal testing. Wide-scale predictive approaches, as described here, answer a direct and current need for generalized prediction frameworks. These can aid in identifying especially sensitive species and toxic chemicals. At the Norwegian Institute for Water Research (NIVA), TERA will be used in this regard and will support several research projects.

In environmental risk assessment it is often uneconomic to assess the hazard and risk a chemical poses to a local species in the environment. These species may not be suitable for lab testing, or may even be endangered and thus are protected by national or international legislation. The currently presented work provides an in silico approach to predict the hazard to such species based on the taxonomic position of the species within the tree of life.

From an economic perspective, TERA and the prediction models are useful tools to evaluate new industrial chemicals during the synthetic in silico stage. Candidate chemicals can be evaluated for their potential environmental hazard, which is in line with the Green Chemistry initiatives by authorities such as the
European Parliament or the US Environmental Protection Agency.

The effect prediction using TERA is also in line with a larger shift in ecological risk assessment towards the use of artificial intelligence \[80\]. We also believe the development of TERA contributes to a methodological change in the community, and encourages others to make their data interoperable.

9.2. TERA as background knowledge

As mentioned, in this work we use TERA directly in prediction models. However, TERA could be used as background knowledge to improve many emerging techniques for toxicity prediction (e.g., \[65\]). These methods often use chemical features, images, fingerprints and so on as input, and machine learning methods such as Convolutional Neural Networks and Random Forests as prediction models \[81, 84\]. These models are often uninterpretable, and the predictions lack domain explanations. TERA can also provide context for machine learning tasks such as pre-processing, feature extraction, transfer and zero/few-shot learning. Furthermore, the knowledge graph is a possible source for the (semantic) explanation of the predictions (e.g., \[43\]).

9.3. Benchmarking KG embedding models

We have shown that embedding TERA brings new challenges to state-of-the-art KGE models with respect to capturing the semantics of the chemicals and the species. Furthermore, as shown in Section 5.4 the sparsity-related measures indicate that TERA represent an interesting KG. KGE models could be benchmarked in a standard KG completion task or in a specific task such as the chemical effect prediction.

9.4. Value to the ontology alignment community

As mentioned in Section 5.4 there does not exist a complete and public alignment between ECOTOX species and the NCBI Taxonomy. Therefore the computed mappings can also be seen as a very relevant resource to the ecotoxicology community. The used alignment techniques achieve high scores for recall over the available (incomplete) reference mappings. However, aligning such large and challenging datasets requires preprocessing before ontology alignment systems can cope with them. We removed all nodes which did not share a word (or shared only a stop word) in labels across the two taxonomies. This quartered the size of ECOTOX and reduced NCBI Taxonomy 50 fold. However, the possible alignment between entities without labels is lost when reducing the dataset size. Thus, the alignment of ECOTOX and NCBI Taxonomy has the potential of becoming a new track of the Ontology Alignment Evaluation Initiative (OAEI) \[51\] to push the limits of large scale ontology alignment tools. Furthermore, the output of the different OAEI participants could be merged into a rich consensus alignment (e.g., as done in the phenotype-disease domain \[28\]) that could become the reference alignment to integrate ECOTOX and NCBI Taxonomy.

9.5. Future work

We plan to extend TERA to include a larger part of ChEBI (which ChEMBL is a part of). ChEBI includes relevant data on the interaction between chemicals and species at a cellular level, which may be very important for chemical effect prediction. In this work we only consider effect data from ECOTOX as this is the largest data set available, however, the inclusion of e.g., TOXCAST \[75\] is in our interest. New sources will always bring more coverage of the domain and will improve TERA for prediction, as background knowledge, and for data access.

We plan to evaluate the effect prediction under different parts of TERA, i.e., which sources in TERA provide value and which do not contribute in terms of the effect prediction. A similar effort in exploring different KG crawling techniques has been explored in \[67\]. In a similar vain, we plan to evaluate how materialization, via OWL reasoning, of TERA’s implicit triples affects prediction performance.

Finally, as mentioned already, some KGE models cannot deal with parts of the structure of TERA. An in-depth analysis of this is an interesting direction for future research. This could be solved by embedding the hierarchy separately, e.g., \[49\], or imposing restrictions on the embeddings, such as a minimum distance constraint.

9.6. Resources

We encourage feedback from domain researchers on extensions to TERA and associated tools.

A snapshot of TERA is available at

https://doi.org/10.5281/zenodo.3559865
This snapshot does not include data that is impractical to re-share (i.e., partial $KG_C$ as described in Section 5). However, we include the full $KG_E$ and $KG_S$.

All the material related to this project is available at https://github.com/NIVA-Knowledge-Graph/.

Source codes to create TERA are available in the TERA GitHub repository. The prediction models and data used for prediction can be found in the KGs_and_Effect_Prediction_2020 GitHub repository. The prediction models require the implementation of the KGE models from the KGE-Keras GitHub repository.

Acknowledgements

This work is supported by the grant 272414 from the Research Council of Norway (RCN), the MixRisk project (Research Council of Norway, project 268294), SIRIUS Centre for Scalable Data Access (Research Council of Norway, project 237889), Samsung Research, Siemens AG, and the EPSRC projects AnaRisk and SNOMED. Health Information Technology Standards. Springer London, 2012.

References

[1] A. Algergawy and M. Samwald. Benchmarking neural embeddings for link prediction in knowledge graphs under semantic and structural changes. J. Web Semant., 64:100590, 2020.

[2] A. Algergawy, M. Cheatham, D. Faria, A. Ferrara, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, N. Karam, A. Khati, P. Lambrix, H. Li, S. Montanelli, H. Paulheim, C. Pesquita, T. Saveta, D. Schmidt, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, O. Zamażal, and L. Zhou. Results of the ontology alignment evaluation initiative 2018. In P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham, and O. Hassanzadeh, editors, Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 26, 2018, volume 2288 of CEUR-WS.org, pages 76–116.

[3] A. Algergawy, D. Faria, A. Ferrara, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, N. Karam, A. Khati, P. Lambrix, H. Li, S. Montanelli, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, O. Zamażal, and L. Zhou. Results of the ontology alignment evaluation initiative 2019. In P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh, and C. Trojahn, editors, Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019, volume 2536 of CEUR Workshop Proceedings, pages 46–85. CEUR-WS.org, 2019.

[4] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vemue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, and J. Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. CoRR, abs/2006.13365, 2020.

[5] M. Alshahrani, M. A. Khan, O. Maddouri, A. R. Kinjo, N. Queralt-Rosinach, and R. Hoehndorf. Neuro-symbolic representation learning on biological knowledge graphs. Bioinform., 33(17):2723–2730, 2017.

[6] H. Arnaout and S. Elbassuoni. Effective searching of rdf knowledge graphs. Journal of Web Semantics, 48:66 – 84, 2018.

[7] T. Benson. Principles of Health Interoperability HL7 and SNOMED. Health Information Technology Standards. Springer London, 2012.

[8] K. Blagec, H. Xu, A. Agibetov, and M. Samwald. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. BMC Bioinformatics, 20(1):178, Apr 2019.

[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD ’08, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery.

[10] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In J. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2787–2795, 2013.

[11] P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. ACM Comput. Surv., 49(2):31:1–31:50, 2016.

[12] A. Breit, S. Ott, A. Agibetov, and M. Samwald. Openbiolink: a benchmarking framework for modeling multi-relational data. In J. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2787–2795, 2013.

[13] J. Chen, P. Hu, E. Jiménez-Ruiz, O. M. Holter, D. Antonyrajah, and I. Horrocks. OWL2Vec*: embedding of OWL ontologies. Mach. Learn., 110(7):1813–1845, 2021.

[14] J. Chen, E. Jiménez-Ruiz, I. Horrocks, D. Antonyrajah, A. Hadian, and J. Lee. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. In European Semantic Web Conference (ESWC), pages 392–408, 2021.

[15] X. Chen, M.-X. Liu, and G.-Y. Yan. Drug–target interaction prediction by random walk on the heterogeneous network. Mol. BioSyst., 8:1970–1978, 2012.

[16] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[17] T. F. Coleman and J. J. Moré. Estimation of sparse jacobian matrices and graph coloring blems. SIAM Journal on Numerical Analysis, 20(1):187–209, 1983.

[18] J. David, J. Euzenat, F. Scharffe, and C. T. dos Santos. The ontology alignment API 4.0. European Semantic Web Conference (ESWC), 2(1):3–10, 2011.

[19] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2d knowledge graph embeddings. In S. A. McIlraith and K. Q. Weinberger, editors, Proceedings of the Thirty-Sev
Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pages 1811–1818. AAAI Press, 2018.

[20] J. A. Doering, S. Lee, K. Kristiansen, L. Evenseth, M. G. Baron, I. Syhte, and C. A. LaLone. In Silico Site-Directed Mutagenesis Infors Species-Specific Predictions of Chemical Susceptibility Derived From the Sequence Alignment to Predict Across Species Susceptibility (SeqAPASS) Tool. Toxicological Sciences, 166(1):131–145, 07 2018.

[21] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Stohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In S. A. Mackassy, C. Perlisch, J. Leskovec, W. Wang, and R. Ghani, editors, The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14, New York, NY, USA - August 24 - 27, 2014, pages 601–610. ACM, 2014.

[22] A. Z. Dudek, T. Arodz, and J. Gámez. Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review. Combinatorial Chemistry & High Throughput Screening, 9(3):213–228, 2006.

[23] J. Euzenat and P. Shvaiko. Ontology Matching, Second Edition. Springer, 2013.

[24] D. Faria, E. Jiménez-Ruiz, C. Pesquita, E. Santos, and F. M. Couto. Towards annotating potential incoherencies in bioprotal mappings. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. A. Knoblock, D. Vrandecic, P. Groth, N. F. Noy, K. Janowicz, and C. A. Goble, editors, The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II, volume 8797 of Lecture Notes in Computer Science, pages 17–32. Springer, 2014.

[25] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The AgreementMakerLight Ontology Matching System. In On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings, pages 527–541, 2013.

[26] J. Fukuchi, A. Kitazawa, K. Hirabayashi, and M. Homma. A practice of expert review by read-across using QSAR Toolbox. Mutagenesis, 34(1):49–54, 01 2019.

[27] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. J. Web Semant., 6(4):309–322, 2008.

[28] I. Harrow, E. Jiménez-Ruiz, A. Splendidani, M. Romacker, P. Woollard, S. Markel, Y. Alam-Faruque, M. Koch, J. Malone, and A. Waaler. Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. J. Biomed. Semant., 8(1):55:1–55:13, 2017.

[29] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. K., V. Mathur, S. Turner, N. Swanston, P. Mendes, and C. Steinbeck. ChEBI in 2016: Improved services and an expanding collection of metabolites. Nucleic acids research, 44(D1):214—9, 2016.

[30] K. Hayashi and M. Shimbo. On the equivalence of holographic and complex embeddings for link prediction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 554–559, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[31] S. R. Heller, A. McNaught, I. V. Pletnev, S. Stein, and D. Tschekhovskoi. Inchi, the IUPAC international chemical identifier. J. Cheminformatics, 7:23:1–23:34, 2015.

[32] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. de Melo, C. Gutiérrez, S. Karrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, and A. Zimmermann. Knowledge graphs. ACM Comput. Surv., 54(4):71:1–71:37, 2021.

[33] E. Jiménez-Ruiz and B. Cuenca Grau. LogMap: Logic-Based and Scalable Ontology Matching. In 10th International Semantic Web Conference (ISWC), pages 273–288, 2011.

[34] E. Jiménez-Ruiz, B. Cuenca Grau, Y. Zhou, and I. Horrocks. Large-scale Interactive Ontology Matching: Algorithms and Implementation. In 20th European Conference on Artificial Intelligence (ECAI), pages 444–449, 2012.

[35] E. Jiménez-Ruiz, B. C. Grau, I. Horrocks, and R. B. Llavori. Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomed. Semant., 2(8-1):S2, 2011.

[36] R. Kadlec, O. Baigar, and J. Kleindienst. Knowledge base completion: Baselines strike back. In P. Blumsom, A. Bordes, K. Cho, S. B. Cohen, C. Dyer, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, and S. Yih, editors, Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP®@ACL, 2017, Vancouver, Canada, August 3, 2017, pages 69–74. Association for Computational Linguistics, 2017.

[37] S. Kim, E. E. Bolton, and S. H. Bryant. Similar compounds versus similar conformers: complementarity between PubChem 2-D and 3-D neighboring sets. Journal of Chemical Information and Modeling, 58(1):82, Nov 2016.

[38] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton. PubChem 2019 update: improved access to chemical data. Nucleic Acids Research, 47(D1):D1102–D1109, 10 2018.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015. Conference Track Proceedings, 2015.

[40] M. Kulmanov, W. Liu-Wei, Y. Yan, and R. Hoehndorf. EL embeddings: Geometric construction of models for the description logic EL++. In S. Kraus, editor, Proceedings of the Twenty-Eight International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pages 6103–6109. ijcai.org, 2019.

[41] C. LaLone, D. Villeneuve, H. Helgen, and G. Ankley. Sequence alignment to predict across-species susceptibility. SE-TAC Europe, Basel, SWITZERLAND, May 11 - 15, 2014.

[42] M. Iare (Skolelaboratoriet i realfag ved Universitetet i Bergen). Smyr i ferskvann. Accessed 11.06.2020.

[43] F. Lécué and J. Wu. Semantic explanations of predictions. CoRR, abs/1805.10587, 2018.

[44] J. Lehmam, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleeft, S. Auer, and C. Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web, 6(2):167–195, 2015.
[45] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady, 10:707, Feb 1966.

[46] X. Liang, D. Li, M. Song, A. Madden, Y. Ding, and Y. Bu. Predicting biomedical relationships using the knowledge and graph embedding cascade model. PLOS ONE, 14(6):1–23, 06 2019.

[47] G. A. Miller. Wordnet: A lexical database for english. Comman. ACM, 38(1):39–41, Nov. 1995.

[48] S. K. Mohamed, V. Nováček, P. Vandenbussche, and E. Muñoz. Loss functions in knowledge graph embedding models. In M. Alam, D. Buscaldi, M. Cochez, F. Osborne, D. R. Recupero, and H. Sack, editors, Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DLAKG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019), volume 2377 of CEUR Workshop Proceedings, pages 1–10. CEUR-WS.org, 2019.

[49] S. Mumtaz and M. Giese. Hierarchy-based Semantic Embeddings for Single-valued & Multi-valued Categorical Variable. Journal of Intelligent Information Systems, 2021. (in press).

[50] E. B. Myklebust, E. Jiménez-Ruiz, J. Chen, R. Wolf, and K. E. Tollefsen. Knowledge graph embedding for ecotoxicological effect prediction. The Semantic Web – ISWC 2019, page 490–506, 2019.

[51] E. B. Myklebust, E. Jiménez-Ruiz, J. Chen, R. Wolf, and K. E. Tollefsen. Ontology alignment in ecotoxicological effect prediction. In 15th International Workshop on Ontology Matching, 2020.

[52] E. B. Myklebust, E. Jimenez-Ruiz, C. Jiaoyan, R. Wolf, and K. E. Tollefsen. Toxicological Effect and Risk Assessment (TERA) Knowledge Graph, November 2020. (Version 1.1.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4244313.

[53] M. Nayeri, C. Xu, Y. Yaghoobzadeh, H. S. Yazdi, and J. Lehmann. Toward understanding the effect of loss function on then performance of knowledge graph embedding, 2019.

[54] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Q. Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In M. A. Walker, H. Ji, and A. Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL-HLT, pages 327–333, 2018.

[55] M. Nickel, L. Rosasco, and T. A. Poggio. Holographic embeddings of knowledge graphs. In D. Schuurmans and M. P. Wellman, editors, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pages 1955–1961. AAAI Press, 2016.

[56] NLM. Medical Subject Headings (MeSH) RDF. 2020. [https://id.nlm.nih.gov/mesh/]

[57] C. S. Parr, N. Wilson, P. Leary, K. Schulz, K. Lanz, L. Walley, J. Hammock, A. Goddard, J. Rice, and M. Studer. The encyclopedia of life v2. Providing global access to knowledge about life on earth, 2014.

[58] C. S. Parr, N. Wilson, P. Leary, K. Schulz, K. Lanz, L. Walley, J. A. Hammock, A. Goddard, J. Rice, M. Studer, J. T. Holmes, and J. Robert J. Corrigan. The Encyclopedia of Life v2: Providing Global Access to Knowledge About Life on Earth. Biodiversity Data Journal, 2:e1079, 2014.

[59] R. Parthasarathi and A. Dhanaw. Chapter 5 - in silico approaches for predictive toxicology. In A. Dhanaw and S. Kwon, editors, In Vitro Toxicology, pages 91 – 109. Academic Press, 2018.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[61] M. A. N. Pour, A. Aleregawy, R. Amini, D. Fariba, I. Fundulaki, I. Harrow, S. Hertling, E. Jimenez-Ruiz, C. Jonquet, N. Karam, A. Khiat, A. Ladhari, P. Lambrix, H. Li, Y. Li, P. Hitzler, H. Paulheim, C. Pesquita, T. Savata, P. Shvaiko, A. Splendi, E. Thieblin, C. Trojahn, J. Vatascinová, B. Yaman, O. Zama, and L. Zhou. Results of the ontology alignment evaluation initiative 2020. In P. Shvaiko, J. Euzenat, E. Jimenez-Ruiz, O. Hassanzadeh, and C. Trojahn, editors, Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020). Virtual conference (originally planned to be in Athens, Greece), November 2, 2020, volume 2788 of CEUR Workshop Proceedings, pages 92–138. CEUR-WS.org, 2020.

[62] J. Pujara, E. Augustine, and L. Getoor. Sparsity and noise: Where knowledge graph embeddings fall short. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1751–1756, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[63] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Meriando. Knowledge graph embedding for link prediction: A comparative analysis. ACM Trans. Knowl. Discov. Data, 15(2):14:1–14:49, 2021.

[64] E. W. Sayers, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, M. Feolo, L. Y. Geer, W. Helmberg, Y. Kapustin, D. Landsman, D. J. Lipman, T. L. Madden, D. R. Maglott, V. Miller, I. Mizrachi, J. Ostell, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, M. Shumway, K. Siroftik, A. Souvorov, G. Starchenko, T. A. Tatussova, L. Wagner, E. Yaschenko, and J. Y. Ye. Database resources of the National Center for Biotechnology Information. Nucleic Acids Research, 37(suppl_1):D5–D15, 10 2008.

[65] A. K. Sharma, G. N. Srivastava, A. Roy, and V. K. Sharma. Toxim: A toxicity prediction tool for small molecules developed using machine learning and chemoinformatics approaches. Frontiers in pharmacology, 8:880–880, Nov 2017.

[66] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng., 25(1):158–176, 2013.

[67] N. P. O. Skrindebakke. Understanding the Role of Background Knowledge in Predictions, 2020. Master’s thesis.

[68] F. Z. Smaili, X. Gao, and R. Hoehndorf. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics, 35(12):2133–2140, 2019.

[69] F. M. Suchaneck, G. Kasnecci, and G. Weikum. Yago: a core of semantic knowledge. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors, Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pages 697–706. ACM, 2007.
Appendix A. Knowledge Graph Embedding Models

In this work, we use 9 KGE models of three major categories: decomposition models, geometric models, and convolutional models. The interested reader please refer to [63] for a comprehensive survey.

A.1. Notation

Throughout this section we use bold letters to denote vectors while matrices are denoted as \( M \). Common notation for all KGE models are, \( \| \cdot \|_p \) for the \( n \)-th norm, \( \langle x, y \rangle \) for the inner product (dot product) between \( x \) and \( y \), \( |x; y| \) is the concatenation of \( x \) and \( y \). \( \text{vec}(X) \) reshapes a matrix into a one-dimensional vector.

The vector representation of an entity and a relation are noted as \( e_x \) and \( e_p \), respectively. These vectors are either in \( \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the embedding dimension.

A.2. Decomposition models

DistMult. Developed by [83] and shown to have state-of-the-art performance on link prediction tasks under optimal hyper-parameters [56]. This model represent the score of a triple as an Hadaman product (dot product) of the vectors representing the subject, predicate, and object of a triple.

\[
S_{\text{DistMult}}(sb,p,ob) = \langle e_{sb}, e_p, e_{ob} \rangle
\]  
(23)

This model does not take the direction of the relation into account, that is, \( S_{\text{DistMult}}(sb,p,ob) = S_{\text{DistMult}}(p,ob,sb) \).

ComplEx. This model use the same scoring function as DistMult [73]. However, the entity vector representation are in the complex space \( e_{sb}, e_p, e_{ob} \in \mathbb{C}^k \) and therefore, the drawback of lacking directionality...
The translational model has the scoring function to be similar to ComplEx \cite{balavzevic2017complex}, and therefore, we expect the performance of RotatE to be improved. HolE has been shown to be equivalent to the object embedding.

\[ SF_{\text{HolE}}(sb, p, ob) = \left( e_{sb} \ast e_{ob} \right) F^{-1} \left[ |F(e_{sb}) \circ F(e_{ob})| \right] \]

where \( F \) and \( F^{-1} \) are the Fourier transform and its inverse, for this model we use \( \pi \) as the elementwise complex conjugate, \( \circ \) denotes the Hadamard product (element-wise). HolE has been shown to be equivalent to ComplEx \cite{balavzevic2017complex}, and therefore, we expect the performance to be similar.

### A.3. Geometric models

**TransE.** The translational model has the scoring function \[ SF_{\text{TransE}}(sb, p, ob) = ||e_{sb} + e_p - e_{ob}||_n. \] Such that if \((sb, p, ob)\) exists in the KG the relational embedding will translate the subject embedding close to the object embedding.

**RotatE.** This model is inspired by Euler’s identity \((e^{i\theta} = \cos(\theta) + i\sin(\theta))\) and scores triples by rotating the relation embedding in complex space. RotatE has been shown to be efficient of modelling symmetric, inverse and composite relations \cite{bordes2013translating}. The scoring function of RotatE is defined as

\[ SF_{\text{RotatE}}(sb, p, ob) = ||e_{sb} \circ e_p - e_{ob}||_n \]

\[ = ||e_{sb} \circ (\cos(\theta_p) + i\sin(\theta_p)) - e_{ob}||_n \]

\[ = ||\Re(e_{sb}) \cos(\theta_p) - \Im(e_{sb}) \sin(\theta_p) - \Re(e_{ob}) \bang\Im(e_{ob}) \bang e_{ob}||_n \]

\[ = ||\Re(e_{sb}) \sin(\theta_p) + \Im(e_{ab}) \cos(\theta_p) - \Im(e_{ob}) \bang e_{ob}||_n. \]

\]

Here, we concatenate the real and complex parts of \(e_{sb} \circ e_p - e_{ob}\). The modulus constraint of \(e_p\) is set equal to 1 and is therefore not included in the scoring function. See the original publication for details of derivation.

**pRotatE.** This model is described as a baseline for RotatE enabling comparison when including modulus information in the model versus limiting to phase information only \cite{bordes2013translating}. pRotatE has the scoring function

\[ SF_{\text{pRotatE}}(sb, p, ob) = 2MC||\sin\left(\frac{\theta_{sb} + \theta_p - \theta_{ob}}{2}\right)||_n \]

where \(\theta = \angle e \) (phase of \(e\)) and \(MC\) is the modulus constraint on \(e_{sb}\) and \(e_{ob}\).

**HAKE.** The hierarchy-aware model use the modulus and the phase part of the embedding vectors \cite{zhang2018hierarchy}. Such that entities at the same level in the hierarchy is modelled using rotation, i.e., phase, and the entities at different levels are modelled using the distance from the origin, i.e., modulus. Therefore, the scoring function of HAKE is modelled in two parts

\[ SF_{\text{HAKE}}(sb, p, ob) = ||e_{sb} \circ e_p - e_{ob}||_n \]

\[ + ||\sin\left(\frac{\theta_{sb} + \theta_p - \theta_{ob}}{2}\right)||_1 \]

where \( \cdot \) is the modulus of \(\cdot\). The authors noted that a mixture bias can be added to \(||e_{sb} \circ e_p - e_{ob}||_n\) to improved performance \cite{zhang2018hierarchy}. We omit these details here.

### A.4. Convolutional models

The final set of models used in this work are convolutional models. We denote convolutions between an image \(X\) and filters \(\omega\) is denoted as \(X \ast \omega\). The models also use dense layers, which is denoted by transform
matrices, e.g., W, note that this also includes bias, even though we do not explicitly state it. Moreover, dropout layers are used between every convolutional and dense layer.

**ConvKB.** The scoring function of ConvKB uses a single convolutional layer and a single dense layer

\[
SF_{\text{ConvKB}}(sb, p, ob) = 
\]  
\[
f(\text{vec}(f([e_{sb}; e_p; e_{ob}]) \ast \omega))W, 
\]

(31)

where vec(x) reshapes x to a 1-dimensional vector. \(\omega\) is the convolution filters. W is the transformation matrix for the output dense layer. ConvKB can easily be extended to use multiple convolution and dense layers.

**ConvE.** In contrast to ConvKB, ConvE only perform convolution over the subject and predicate image (concatenated and reshaped) and multiplies the output dense layer with the object vector as such

\[
SF_{\text{ConvE}}(sb, p, ob) = 
\]  
\[
f(\text{vec}(f([e_{sb}; e_p]) \ast \omega))W_{ob} \text{vec}(X), 
\]

(32)

where X reshapes x into a 2-dimensional image. Here, the last dimension of W is equal to the embedding dimension. This model can also be extended with multiple convolution and dense layers, however, found that this did not yield improved results.

A.5. Loss functions

Work on KGE models usually define loss functions specific to the models. However, as shown in 48, 53 the choice of loss function has a huge impact on model performance. In this work we use four loss functions. We experimented with other loss functions, e.g., absolute/square error, however, these did not materialize in improved results.

To optimize a loss function we need to generate negative examples. Under the local closed world assumption we replace the object of each true triple with negative examples. Under the local closed world assumption we replace the object of each true triple with negative examples. Under the local closed world assumption, we replace the object of each true triple with negative examples. Under the local closed world assumption, which corrupt both the subject and the object of true triples (illustrated by Fig. 3 in 48). This model can also be extended with multiple convolution and dense layers, however, \[19\] found that this did not yield improved results.

The number of negative samples sampled per positive sample is controlled by a hyper-parameter. However, 56 show that the largest possible number is favorable.

**Pointwise hinge.** The objective of pointwise losses minimize the scores of negative triples and maximize the score of positive triples.

\[
L_{H_1} = \sum_{t \in X} [\gamma - y_t S(t)]_+ 
\]

(33)

where X is the set of positive and negative triples, \(y\) is the triple label (−1 for false and 1 for true) and \(S(t)\) is the score of triple \(t\). \(\gamma\) is the margin hyper-parameter. \([x]_+\) is the positive part of \(x\).

**Pointwise logistic.** In contrast to hinge loss, logistic loss applies a larger non-linear loss to predictions that are further away from the true label.

\[
L_{L_1} = \sum_{t \in X} \log(1 + \exp(-y_t S(t))) 
\]

(34)

**Pairwise hinge.** The objective of pairwise loss functions is to maximize the distance (in score) between a positive and a negative triple.

\[
L_{H_2} = \sum_{t' \in X^+} \sum_{t \in X^-} [\gamma + S(x^-) - S(x^+)]_+ 
\]

(35)

where \(X^+\) and \(X^-\) are the sets of positive and negative triples, respectively. \(\gamma\) is the margin hyper-parameter, which for pairwise hinge loss represents the maximum score discrepancy between a positive and negative score.

**Pairwise logistic.** Akin to the move from pointwise to pairwise hinge, pairwise logistic maximizes the distance between positive and negative triples, however, in a non-linear way

\[
L_{L_2} = \sum_{t' \in X^+} \sum_{t \in X^-} \log(1 + \exp(S(x^-) - S(x^+)) 
\]

(36)

A.6. Implementation

We have implemented the KGE models in Keras and the model codes are available at https://github.com/NIVA-Knowledge-Graph/KGE-Keras. This enables us to easily use the KGE models as components in other models as described in Section 6.