Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings

Erik B. Myklebust\textsuperscript{a,b,f,*}, Ernesto Jiménez-Ruiz\textsuperscript{c,b,**}, Jiaoyan Chen\textsuperscript{d}, Raoul Wolf\textsuperscript{a,g} and Knut Erik Tollefsen\textsuperscript{a,e}

\textsuperscript{a}Norwegian Institute for Water Research, Oslo, Norway
\textsuperscript{b}SIRIUS, University of Oslo, Oslo, Norway
\textsuperscript{c}City, University of London, London, United Kingdom
\textsuperscript{d}University of Oxford, Oxford, United Kingdom
\textsuperscript{e}Norwegian Institute for Water Research, Oslo, Norway
\textsuperscript{f}NORSAR, Kjeller, Norway
\textsuperscript{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 [6] 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 semantic 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 contain 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
2.2. Ontology-enhanced knowledge graphs

In this work we consider the most broadly accepted notion of knowledge graph within the Semantic Web: an ontology enhanced RDF-based knowledge graph (KG) [32]. This kind of knowledge graph enables the use of the available Semantic Web infrastructure, including SPARQL engines and OWL reasoners.\(^2\)

Thus, in our setting, KGs are composed by RDF triples in the form of \(\langle sb, p, ob \rangle \in E \times R \times (E \cup L)\), 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).

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.\(^3\)

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 13 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

\(^1\)Not to be confused with SPARQL endpoint.

---

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>KG</td>
<td>Knowledge graph</td>
</tr>
<tr>
<td>KGE</td>
<td>Knowledge graph embedding</td>
</tr>
<tr>
<td>(t)</td>
<td>A triple</td>
</tr>
<tr>
<td>(sb)</td>
<td>The subject of a triple</td>
</tr>
<tr>
<td>(ob)</td>
<td>The object of a triple</td>
</tr>
<tr>
<td>(p, r)</td>
<td>The predicate/relation of a triple</td>
</tr>
<tr>
<td>(E)</td>
<td>The set of KG triples</td>
</tr>
<tr>
<td>(R)</td>
<td>The set of KG entities</td>
</tr>
<tr>
<td>(L)</td>
<td>The set of literal values</td>
</tr>
<tr>
<td>(c)</td>
<td>A chemical</td>
</tr>
<tr>
<td>(S)</td>
<td>Refers to species</td>
</tr>
<tr>
<td>(C)</td>
<td>Refers to chemicals</td>
</tr>
<tr>
<td>(k)</td>
<td>Chemical concentration</td>
</tr>
</tbody>
</table>

| Table 1 | Key symbols and acronyms used throughout the paper. |

\(taxon\). \textit{Species} is a taxon which is also a leaf node in the taxonomy. An \textit{Organism} denotes an individual living organism which is an instance of a species. \textit{Chemicals} or compounds are unique isotopes of substances consisting of two or more atoms. \textit{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. \textit{Endpoint}\(^1\) denotes a measured effect on the test population at a certain time; e.g., lethal concentration to 50% of test population (LC50) measured at 48 hours. Note that, an experiment can have several endpoints, e.g., LC50 at 48 hours and LC100 at 96 hours (lethal concentration for all test organisms). See Table 2 for the most common endpoints.

\(^2\)RDF, RDFS, OWL and SPARQL are standards defined by the W3C: https://www.w3.org/standards/semanticweb/

\(^3\)\(E\) is the set of all classes and instances, \(R\) is the set of all properties, while \(L\) represents the set of all literal values.

\(^4\)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/
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]).

The embeddings of the entities in a KG are commonly learned by (i) defining a scoring function over a
triplet, which is typically proportional to the probability
of the existence of that triple in the KG.5 i.e., \( SF : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}, SF \propto P(\langle sb, p, ob \rangle \in KG) \); and (ii) mini-
mizing a loss function (i.e., deviation of the prediction
of the scoring function with respect to the truth avail-
able in the KG). More specifically, KGE models (i) ini-
tialize the entities in a triple \( \langle sb, p, ob \rangle \) into a vector
representation \( e_{sb}, e_{p}, e_{ob} \in \mathbb{R}^k \) or \( \mathbb{C}^k \), where \( k \) is the di-
mension 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.6 The decomposi-
tion 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 func-
tions 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 con-
text, risk is the result of the intrinsic hazards of a sub-
stance on species, populations or ecosystems, com-
bined 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 envi-
ronmental concentrations of one or more chemicals, while
effects (hazards) are characterized for a number of

---

5For the embedding process, we focus on triples where \( o \in \mathcal{E} \) is a
class or an instance.

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

---

species in the laboratory as a proxy for more ecolog-
ically 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 chemi-
ical or the mixture of many chemicals is used to identify
chemicals with the highest RQs (risk drivers), ident-
ify relevant modes of action7 (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 ver-

tion of the one in use at the Norwegian Institute for
Water Research,8 however, similar methodologies are
used across regulatory risk assessment pipelines.

The chemical effect data is gathered during labora-

tory 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 ex-
posure. 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 accu-
rate depiction of the long term risk to an ecosys-

tem. The data must cover the relevant chemicals and
species present in the ecosystem, e.g., an ecologi-
cal risk assessment of agricultural runoff in Norway
will mostly concern pesticides and waterflees, cope-
pods, and frogs, among other species [42]. Just with
a few relevant chemicals and species the search space

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

8NIVA: https://www.niva.no/en
becomes immense and performing laboratory experiments becomes unfeasible. Thus, it is essential to develop \textit{in silico} methods to extrapolate new chemical-species effects from known combinations. We differentiate among two types complementary strategies: \textit{(i)} highly specialized (restricted in chemical and species domains) models to predict chemical concentrations that will have an effect on a test species, and \textit{(ii)} models that produce rankings of highly representative chemical-species pair hypothesis which can be 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, \textit{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 (\textit{cf.} reviews \cite{22, 26}). This relationship is described using derived chemical features. Some features are simple, \textit{e.g.}, octanol-water partition coefficient or logP, others concern the entire chemical, \textit{e.g.}, chemical fingerprints. The basis of the QSAR relationship is usually modeled as polynomial equations. Parthasarathi and Dhawan \cite{59} take this further by using the logarithm of chemical concentration to achieve a polynomial relationship: \( \log(1/\kappa) = f(\pi) + g(\sigma) \), \( f \in P_2 \) 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\(^9\) 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 \cite{20, 41}. SeqAPASS uses a large amount of data available for humans, mice, rats, and zebrafish to extrapolate to areas with lower coverage.

\(^9\)Measure of the absence of attraction to water.

---

**Table 2**

The most frequent endpoints in ECOTOX \cite{74} chemical effect data.

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Frequency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>0.21</td>
<td>Not reported</td>
</tr>
<tr>
<td>NOEL</td>
<td>0.17</td>
<td>No-observable-effect-level</td>
</tr>
<tr>
<td>LC50</td>
<td>0.16</td>
<td>Lethal concentration for 50% of test population</td>
</tr>
<tr>
<td>LOEL</td>
<td>0.14</td>
<td>Lowest-observable-effect-level</td>
</tr>
<tr>
<td>NOEC</td>
<td>0.05</td>
<td>No-observable-effect-concentration</td>
</tr>
<tr>
<td>EC50</td>
<td>0.05</td>
<td>Effective concentration for 50% of test population</td>
</tr>
<tr>
<td>LOEC</td>
<td>0.04</td>
<td>Lowest observable effect concentration</td>
</tr>
<tr>
<td>BCF</td>
<td>0.03</td>
<td>Bioconcentration factor</td>
</tr>
<tr>
<td>NR-LETH</td>
<td>0.02</td>
<td>Lethal to 100% of test population</td>
</tr>
<tr>
<td>LD50</td>
<td>0.02</td>
<td>Lethal dose for 50% of test population</td>
</tr>
<tr>
<td>Other</td>
<td>0.11</td>
<td></td>
</tr>
</tbody>
</table>
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 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 [86].

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, 46].

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. The initial inspiration for TERA was the aid of ecological risk assessment (see Section 5.3). However, by integrating these sources into a KG, we were also able to directly apply TERA into the prediction

---

10 Resources to create and access TERA: https://github.com/NIVA-Knowledge-Graph/TERA
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: PubChem, ChEMBL, MeSH, and Wikidata mappings (~110M compounds); and (iii) biological effects: ECOTOXicology Knowledgebase (~1M results, ~12k compounds, ~13k species), and system-generated mappings. These three distinct parts make up the sub-KGs of TERA, i.e., (i) the Taxonomy sub-KG (KG\textsubscript{T}), (ii) the Chemical sub-KG (KG\textsubscript{C}), and (iii) the Effects sub-KG (KG\textsubscript{E}). 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.\[11\] 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

\[13\]ftp://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBL-RDF/
\[14\]https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020
\[15\]Prefixes associated to the URI namespaces of entities in TERA: et: (ECOTOXicology knowledgebase), ncbi: (NCBI taxonomy), eol: (Encyclopedia of Life), mesh: (Medical Subject Heading), compound: (PubChem compound), descr: (PubChem descriptors), vocab: (PubChem vocabulary), inchikey: (InChIKey identifiers), env: (Environment Ontology), cheminf: (Chemical information ontology), chembli: (ChEMBL), chembli_m: (ChEMBL molecule subset), chembli_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.
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 ECOTOXicology 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 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 [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 [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 (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 [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 a 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) [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.

---


17While InChI is unique, InChIKey is not, and collisions have greater than zero probability [79].

18In the context of the paper “taxonomy” typically refers to a classification of organisms.
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 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 $t_1$–$t_{12}$). 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\footnote{As defined by U.S. EPA. Note that species hierarchies are contested among researchers.} 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\footnote{QUDT 1.1: http://linkedmodel.org/catalog/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 the NCBI Taxonomy species.\footnote{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.} We have used three methods, two state-of-art ontology alignment systems and a baseline, to align ECOTOX and the NCBI Taxonomy: (i) LogMap \citep{33, 34}, (ii) AgreementMakerLight (AML) \cite{25}, and (iii) a string matching algorithm based on Levenshtein distance \cite{45}. LogMap and AML were chosen since they have performed well across many datasets in the Ontology Alignment Evaluation Initiative (e.g., \cite{2, 3, 61}). Most mappings in our setting are expected to be lexical, therefore, we also selected a purely lexical matcher to 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
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, 8CHRNB4, 9DETA-20, 10DETA Epichlorohydrin, 11Has component, 12Triclocarban, 13Trichlorocarbanilide-containing product, 14Similar to, 153-Chloromethyl-N,N-diethylbenzamide.
Fig. 3. Example of an ECOTOX test and related triples.

@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> .

et:MilligramPerLiter
  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.

<table>
<thead>
<tr>
<th>Method</th>
<th>1-to-1 mappings</th>
<th># M</th>
<th>R</th>
<th>P^≈</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogMap</td>
<td></td>
<td>20.585</td>
<td>0.81</td>
<td>0.87</td>
</tr>
<tr>
<td>AML</td>
<td></td>
<td>14.148</td>
<td>0.77</td>
<td>0.94</td>
</tr>
<tr>
<td>String similarity (≥ 0.8)</td>
<td></td>
<td>20.423</td>
<td>0.76</td>
<td>0.87</td>
</tr>
<tr>
<td>Consensus (LogMap ∩ AML)</td>
<td></td>
<td>12.740</td>
<td>0.76</td>
<td>0.98</td>
</tr>
<tr>
<td>LogMap ∪ AML</td>
<td></td>
<td>21.145</td>
<td>0.83</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Table 6

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

filtered according to the system computed confidence.

A partial mapping curated by experts can be obtained through the ECOTOX Web.\(^{22}\) 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

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

\[
M^≈ = \{(e_e, owl:sameAs, e_n) \in M | e_e \in E^e_{ref} \land e_n \in E^e_{ref}\},
\]

where \(M_{ref}\) is the (incomplete) reference mapping set and \(M\) is the set of generated mappings between entities \(e_e \in E^e\) from ECOTOX and entities \(e_n \in E^n\) from the NCBI Taxonomy, \(E^e_{ref} \subseteq E^e\) and \(E^n_{ref} \subseteq E^n\) are the sets of entities that appear in the reference mappings. Thus, \(M^≈\) 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 \(M^≈\).

We have selected the union of the 1-to-1 equivalence\(^{23}\) mappings computed by AML and LogMap to be integrated within TERA, as they represent the map-

\(^{22}\)ECOTOX interface: https://cfpub.epa.gov/ecotox/search.cfm

\(^{23}\)There is no need for more complex mappings in this use case.
We useWikidataas 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 in large parts manually curated 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 the columns of 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, ChEMBL, 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 Table 5.

These mappings are not complete, but for some the coverage is large. 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 ($KG_3$) 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 $nodes.dmp$. 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 $t_{21}$-$t_{22}$ and $t_{24}$-$t_{25}$ in Table 5.

(ii) To aid alignment between the NCBI Taxonomy and the ECOTOX identifiers, we add the synonyms found in $names.dmp$. Here, the taxon identifier, 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 $t_{29}$ 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 ($KG_2$) is created from PubChem [38], ChEMBL [29], and MeSH [56]. 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.

---

28Wikidata endpoint: https://query.wikidata.org/sparql
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 {?taxon owl:sameAs ?ncbi , ?eol .}
WHERE {
  OPTIONAL {
  }
  OPTIONAL {
    BIND(IRI(CONCAT("https://eol.org/pages/",?eol_id)) AS ?eol)
  }
}

Listing 2: Construct taxon mapping between Wikidata and NCBI and EOL. wd:Q16521 is the class of all taxa, while wdt:P31, wdt:P685 and wdt:P830 are the relations instance of, NCBI Taxonomy ID and Encyclopedia of Life ID, 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 t35, t36 and t37 in Table 5.

Parent chemical data in PubChem is limited to permutations e.g., bonds, polarity, and part of mixtures axioms (triple t34 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 $\geq 0.9$, see triple t38 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 t32 in Table 5). These targets are organized in a hierarchy using chembl:relSubsetOf relations (see triple t33). 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 t30 and t31 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

$^{25}$ Default value used in PubChem [37].
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>

WHERE {
  ?chemical wdt:P31 wd:Q11173 .
  OPTIONAL {
    BIND(IRI("https://cfpub.epa.gov/ecotox/chemical/",
              REPLACE(?cas_id,'-',''))) AS ?cas
  }
  OPTIONAL {
    BIND(IRI("http://rdf.ebi.ac.uk/resource/chembl/molecule/",
               ?chembl_id)) AS ?chembl
  }
  OPTIONAL {
    BIND(IRI("http://id.nlm.nih.gov/mesh/",?mesh_id)) AS ?mesh
  }
  OPTIONAL {
               ?pubchem_id)) AS ?pubchem
  }
  OPTIONAL {
               ?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.

and accessibility of knowledge enables optimal decision making.

The knowledge in TERA can be accessed via predefined queries (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.

26Predefined queries are typically abstractions of SPARQL queries.
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX et: <https://cfpub.epa.gov/ecotox/> .
PREFIX et_endpoint: <https://cfpub.epa.gov/ecotox/endpoint/> .
PREFIX et_effect: <https://cfpub.epa.gov/ecotox/effect/> .
PREFIX qudt: <http://qudt.org/schema/qudt#> .

WHERE {
  _:b a et:Test ;
  et:species ?s .
  et:chemical ?c .
  et:hasResult [ et:endpoint et_effect:Mortality ; et:concentration [
  ]
}

Listing 4: Query to select all species, chemicals, concentrations and units, where the species is endemic to the Oslofjord.

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\text{S}) and the Chemical sub-KG (KG\text{C}), respectively, are embedded into low-dimensional vectors; while the Effects sub-KG (KG\text{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\textsuperscript{27} and TERA’s KG\text{C} and KG\text{S} (triples involving literals are removed). We follow Pujara et al. [62] and calculate the relational density, \(RD = |T|/|R|\), and entity density, \(ED = 2|T|/|\mathcal{E}|\), where \(T\), \(R\), and \(\mathcal{E}\) are the sets of triples, relations, and entities in the knowledge graph, respectively. The entity entropy (EE) and the relation entropy (RE) indicate whether there are biases (the lower EE or RE, the larger bias) in the triples in the KG [62], and are calculated as

\[
P(r) = \frac{|t.p = r|}{|T|},
\]

(4)

\[
P(e) = \frac{|t.sb = e| + |t.ob = e|}{|T|},
\]

(5)

\textsuperscript{27}YAGO3-10 [69], FB15k-237 [9], WN18 [47] and WN18RR [19].
\[ RE = \sum_{r \in R} -P(r)\log(P(r)), \quad (6) \]
\[ EE = \sum_{e \in E} -P(e)\log(P(e)), \quad (7) \]

where \(|r_p = r|\) is the number of triples with \(r\) as predicate, and \(|r_{sb} = e| + |r_{ob} = e|\) is the number triples with \(e\) as subject or object.

In addition, we calculate the absolute density of the graph, which is \(AD = |T|/(|E|(|E| - 1))\). 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 a suitable background knowledge to extrapolate effect data and, at the same time, an interesting dataset to benchmark state-of-the-art knowledge graph embedding models. Note that using the full TERA (i.e., \(KG\_C\) and \(KG\_S\)), according to RD, will be more challenging than using the reduced TERA fragments (i.e., \(KG\_C\) and \(KG\_S\)) for prediction. Full details of the construction of \(KG\_C\) and \(KG\_S\) are given in Section 7.1.1.

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 adverse biological effect of chemicals on species: (i) a multilayer perceptron (MLP) model (our baseline), (ii) the baseline model fed with pre-trained KG embeddings, (iii) a model that simultaneously trains the baseline model and the KGE models (i.e., it fine-tunes the KG embeddings). A MLP was chosen as baseline as it is a basic model where additional components and penalties can be easily added and assessed as we do in our third model (see Section 6.3).

The models have three inputs, namely a chemical \(c\), a species \(s\), and a chemical concentration \(\kappa\) (denoted \(x_{c,s,\kappa}\)). The output is a binary value that represents whether the chemical at the given concentration has a lethal effect on the species:

\[ y_{c,s,\kappa} = \begin{cases} 1 & \text{if } c \text{ is lethal to } s \text{ at } \kappa, \\ 0 & \text{otherwise.} \end{cases} \quad (8) \]

Note that the effect can have a more fine-grained categorization (endpoints LC\(_x\), LD\(_x\), EC\(_x\)), and NR-LETH in Table 2). Without losing the generality in introducing and evaluating our effect prediction methods, we simplify the effect into two cases: “lethal” and “non-lethal”.

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 \(e\) and \(e_p\), 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. \(n_h\), hidden layers are appended to the embedding \(e\) of the chemical \(c\), \(n_s\), hidden layers are appended to the embedding \(e_s\) of species \(s\), and \(n_k\) 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):

\[ y_c^0 = e_c, y_s^0 = e_s, y_\kappa^0 = \kappa \quad (9) \]
\[ y_c^h = \text{ReLu}(y_c^{h-1}W_c^h + b_c^h), h \in \{0, \ldots, n_c\} \quad (10) \]
\[ y_s^h = \text{ReLu}(y_s^{h-1}W_s^h + b_s^h), h \in \{0, \ldots, n_s\} \quad (11) \]
\[ y_\kappa^h = \text{ReLu}(y_\kappa^{h-1}W_\kappa^h + b_\kappa^h), h \in \{0, \ldots, n_k\} \quad (12) \]
\[ y^0 = [y_c^0, y_s^0, y_\kappa^0] \quad (13) \]
\[ y^h = \text{ReLU}(y^{h-1}W^h + b^h), h \in \{1, \ldots, n\} \quad (14) \]
\[ \hat{y} = \sigma(y^nW^n + b^n) \quad (15) \]

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

\[ e_c = \delta_cW_c, \quad e_s = \delta_sW_s \quad (16) \]

where \(\delta_c\) and \(\delta_s\) denote the one-hot encoding vectors of the chemical entity \(c\) (w.r.t. all the entities in \(E_C\) from \(KG\_C\)) and the species entity \(s\) (w.r.t. all the en-

\[ ^{28}\text{If effect is mortality (e.g., see Table 4).} \]
Fully connected layers

Output layer

(a) Simple setting. Without transformation layers: \( n_c = 0, n_s = 0, n_K = 0 \) and \( n = 1 \).

(b) Complex setting. Model with branches/transformation layers. In contrast to the simple setting, here \( n_c \gg 1, n_s \gg 1, n_K \gg 1 \) and \( n \gg 1 \).

Fig. 4. Baseline model. Inputs: \( c, s, K \) as in Equation (9); Outputs: \( \hat{y} \) as in Equation (15).

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\( C \) and the species KG\( S \).

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 \( W_c \) and \( W_s \) in Equation (16) by the matrices composed of embeddings by the respective KGE models. Namely \( W_c \) is set to \( [e_{c,1}; e_{c,2}; \ldots; e_{c,|E_c|}] \), \( W_s \) is set to \( [e_{s,1}; e_{s,2}; \ldots; e_{s,|E_s|}] \), where \([::]\) denotes stacking vectors, \( e_{c,i} \) denotes the embedding of the \( i \)th chemical in the chemicals KG\( C \), \( e_{s,i} \) denotes the embedding of the \( i \)th species in the species KG\( S \).

In the experiments we refer to these models as Simple PT KG\( C \)-KG\( S \) and Complex PT KG\( C \)-KG\( S \), depending on the selected MLP setting, where PT stands for pre-trained, and KG\( C \) 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 \( W_C \) and \( W_S \), and the MLP weights (\( W_b \) 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

\[ L = \alpha_C L_{KGEC} + \alpha_S L_{KGES} + \alpha_{MLP} L_{MLP} \]  \hspace{1cm} (17)\]

where \( L_{KGEC} \) and \( L_{KGES} \) 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. \( L_{MLP} \) is calculated as

\[ L_{MLP} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \]  \hspace{1cm} (18)\]

where \( N \) denotes the size of 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 KGs 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.

\(^{30}\) Appendix A.5 introduces the used loss-functions in this work.

\(^{31}\) Section 7.1 describes how the known effect data extracted from ECOTOX is split into training, validation and test sets.
These steps are repeated until the loss (only $L_{\text{MLP}}$) over the validation set stops improving.

In the experiments we refer to these models as Simple FT $\text{KGE}_C$-$\text{KGE}_E$ and Complex FT $\text{KGE}_C$-$\text{KGE}_E$, depending on the selected MLP setting, where FT stands for fine-tuning, and $\text{KGE}_C$ and $\text{KGE}_E$ 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.\textsuperscript{32}

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.\textsuperscript{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 $\text{KG}_E$ is limited to four features, namely, chemical, species, chemical concentration, and effect. The chemical concentrations ($\kappa$, 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 does not cause death in the test species. We label lethal effects as 1 and non-lethal effects as 0.

(ii) $\text{KG}_C$. For each chemical in the effect data, we extract all triples connected to them using a directed crawl. This reduces the size of $\text{KG}_C$ 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 unpractical 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) $\text{KG}_S$. The same steps as for $\text{KG}_C$ 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 $\text{KG}_C$ and $\text{KG}_S$ are primarily hierarchical and we start the crawls at the leaf nodes.

These steps reduce $\text{KG}_C$ to 241,442 triples and $\text{KG}_S$ to 59,673 triples. Some statistics of $\text{KG}_C$ and $\text{KG}_S$, and the reduced fragments $\text{KG}'_C$ and $\text{KG}'_S$, are given in Table 7 (Section 5.4). In the rest of the paper we refer to TERA’s reduced sub-KGs simply as $\text{KG}_C$ and $\text{KG}_S$.

The transformation from TERA’s $\text{KG}_C$ and $\text{KG}_S$ 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 $\mathbf{W}_c$ or $\mathbf{W}_g$ in Equation (16), depending on which sub-KG is transformed.\textsuperscript{34} 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.

\textsuperscript{32}https://github.com/NIVA-Knowledge-Graph/KGs_and_Effect_Prediction_2020

\textsuperscript{33}All data used to create TERA was downloaded on the 14th of May 2020.

\textsuperscript{34}$i \in [0, |\mathcal{E}_c| - 1]$ for $\text{KG}_C$ and $i \in [0, |\mathcal{E}_s| - 1]$ for $\text{KG}_S$.
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.

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, $RL_{ep} = L_{ep} / L_0$, 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. 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 $KG5$. As shown in Section 5.4, the major difference between $KGC$ and $KG5$ 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-$|\mathcal{E}|$ scoring, where $|\mathcal{E}|$ 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 vari-

<table>
<thead>
<tr>
<th>KGE hyper-parameters</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss function</td>
<td>${L_H, L_{LH}, L_{KL}, L_{L2}}$</td>
</tr>
<tr>
<td>Margin (only hinge loss)</td>
<td>${1, 2, \ldots, 10}$</td>
</tr>
<tr>
<td>Bias (only geometric models)</td>
<td>${0, 1, \ldots, 20}$</td>
</tr>
<tr>
<td>Embedding dimension</td>
<td>${100, 101, \ldots, 400}$</td>
</tr>
<tr>
<td>Negative samples</td>
<td>${10, 11, \ldots, 100}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prediction hyper-parameters</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_c$ (10), $n_s$ (11), $n_k$ (12), $n$ (14)</td>
<td>${0, 1, 2, 3}$</td>
</tr>
<tr>
<td># units (10), (11), (14)</td>
<td>${2^a \text{ with } a \in {4, 5, \ldots, 10}}$</td>
</tr>
<tr>
<td># units (12)</td>
<td>${2^a \text{ with } a \in {2, 3, 4, 5}}$</td>
</tr>
</tbody>
</table>

Table 8

Hyper-parameter choices for the models. Please refer to the Equations (9)-(15) in Section 6.1 for the prediction hyper-parameters.
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 KGE embeddings learned with the respective KGE models. For example, the Complex PT DistMult-HAKE model is initialized with the learned weights with the Complex PT DistMult-HAKE model and the pre-trained KGE 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 6.1, we use two settings in our classification models: simple and complex. This will help us isolate the effects of the KGE 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). The same can be seen for PT HAKE-DisMult from strategy (iii) to (iv), where the number of layers increases. Overall we can see that the layer configurations of the chemical branch is more com-

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss function</th>
<th>Margin</th>
<th>Bias</th>
<th>Embedding dimension</th>
<th>Negative Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistMult</td>
<td>$L_{d_2} / L_{d_1}$</td>
<td>-/1/2</td>
<td>-</td>
<td>143 / 383</td>
<td>28 / 43</td>
</tr>
<tr>
<td>ComplEx</td>
<td>$L_{d_2} / L_{d_1}$</td>
<td>-/1/4</td>
<td>-</td>
<td>163 / 372</td>
<td>27 / 42</td>
</tr>
<tr>
<td>HolE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>6/-</td>
<td>-</td>
<td>188 / 376</td>
<td>30 / 100</td>
</tr>
<tr>
<td>TransE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>4/7</td>
<td>14/20</td>
<td>226 / 196</td>
<td>23 / 57</td>
</tr>
<tr>
<td>RotatE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>5/2</td>
<td>16/6</td>
<td>271 / 398</td>
<td>75 / 22</td>
</tr>
<tr>
<td>pRotatE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>-/-</td>
<td>14/16</td>
<td>164 / 210</td>
<td>34 / 82</td>
</tr>
<tr>
<td>HAKE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>-/-</td>
<td>12/10</td>
<td>108 / 359</td>
<td>56 / 13</td>
</tr>
<tr>
<td>ConvKB</td>
<td>$L_{d_2} / L_{d_1}$</td>
<td>-/1/5</td>
<td>-</td>
<td>248 / 276</td>
<td>18 / 90</td>
</tr>
<tr>
<td>ConvE</td>
<td>$L_{h_2} / L_{h_1}$</td>
<td>7/3</td>
<td>-</td>
<td>228 / 196</td>
<td>68 / 40</td>
</tr>
</tbody>
</table>

Table 9

Best hyper-parameters for KGE models. The two values before and after / are for the embeddings of $KG_C$ and $KG_S$, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>Sampling</th>
<th># units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex one-hot</td>
<td>(i)</td>
<td>$(128)/(128)/-/-$</td>
</tr>
<tr>
<td>Complex PT DistMult-HAKE (top-1 in (ii))</td>
<td>(ii)</td>
<td>$(128)/(256)/(8,8)/-$</td>
</tr>
<tr>
<td>Complex PT HolE-ConvKB (top-1 in (ii))</td>
<td>(iii)</td>
<td>$(256,128)/(128)/(4,4,4)/-$</td>
</tr>
<tr>
<td>Complex PT HAKE-DistMult (top-1 in (iii,iv))</td>
<td>(iv)</td>
<td>$(256,256)/(128)/(8,8)/(128)$</td>
</tr>
</tbody>
</table>

Table 10

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: $(b_1, b_2, ..., b_n)/((b_1, ..., b_n)/((b_1, ..., b_n))/((b_1, ..., b_n))/((b_1, ..., b_n))$ as in Equations (10), (11), (12), and (14). $|$ denotes no hidden layers.

e.g., $(128)/(256)/(8,8)/-$ denotes $n_c = 1, n_s = 1, n_k = 2, n = 0$ and $b_1 = 128, b_2 = 256, b_3 = 8$ and $b_4 = 8$.

able, 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 need 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 KGE embeddings learned with the respective KGE models. For example, the Complex PT DistMult-HAKE model is initialized with the learned weights with the Complex PT DistMult-HAKE model and the pre-trained KGE 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.
7.2. Prediction results

In this section we present a summary of the conducted chemical effect prediction evaluation. Complete results are available at the project repository. 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 \( \kappa \). We use several metrics to compare the different prediction models. These are Sensitivity \((i.e., \text{recall})\), Specificity, and Youden’s index \((YI)\) [85]. 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 \((i.e., \text{harmful chemicals})\) than negative samples \((i.e., \text{non-harmful chemicals})\) 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

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

\(i.e.,\) 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 \((\text{ROC})\) curve over a random model and can be used to select the optimal decision threshold in a production environment \((\text{based on validation data})\). We do not present ROC \((\text{or area under ROC, \text{AUC}})\) as a metric as it correlates \((\geq 0.99)\) 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 \((\text{or in our case, a toxicity test})\). 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 \((\text{based on} YI)\) for the baseline model using one-hot and pre-trained \((\text{PT})\) KG embeddings, and the fine-tuning \((\text{FT})\) models using the same combination of KGE models as the selected PT-based models. We have also included a model with middling performance \((i.e., 40 \text{ out of } 81 \text{ models})\) and the worst performing model. Note that for the PT- and FT-based models we have evaluated 81 combinations \(\text{KGE}_{\text{S}}-\text{KGE}_{\text{S}}\) of KGE models. All models were evaluated using the simple and complex MLP settings. For example, the model \text{Complex FT DistMult-HoIE} denoted that fine-tuning was used together with the complex MLP setting, and DistMult was selected to embed the chemicals \(\text{KG}_{\text{C}}\) while HoIE was used to embed the species \(\text{KG}_{\text{S}}\). 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 \textbf{bold} are the best mean results of the corresponding metrics. Underlined results are where there is a \( \geq 32\% \) chance that a single run outperforms the best mean \((i.e., \text{one standard deviation contains } 68\% \text{ of results, assuming normally distribute results})\). 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 $Y_I$ as the best simple pre-trained model. The 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, especially 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 pRotatE-ConvE and Complex PT 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 $Y_{I_{max}}$. 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 deviation 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 $Y_I$ and $Y_{I_{max}}$ 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 $Y_I = 0.021$ to $Y_I = 0.726$ 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 KGE 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 15 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 15 we can see that the KGE models used to embed the chemicals $KC$ in the best performing models is distributed evenly across most models and settings. This indicates that the performance of

<table>
<thead>
<tr>
<th>Model</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>$Y_I$</th>
<th>$Y_{I_{max}}$</th>
<th>$\sigma_{max}$</th>
</tr>
</thead>
</table>
the prediction models is not highly correlated with the use of a KGE model on \( K_{GC} \). Referencing Table 7, the high relational density in \( K_{GC} \) can contribute to worse performance [62] and therefore equal distribution of models in Table 15. This is different for \( K_{GS} \). For sampling strategies (i) and (ii), HAKE is extensively used in the top models to embed \( K_{GS} \). HAKE is designed to embed hierarchies. Therefore, this indicates that in strategies (i) and (ii) the hierarchical structure of \( K_{GS} \) dwarfs the rest of the KG. \( K_{GS} \) has a higher entity density and lower entity entropy (Table 7) than \( K_{GC} \). 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 \( K_{GS} \), 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.38 In Figure 6, we present how the \( YI \) metric depends on the explained variance of the top-10 principal components (i.e., \( \sum_{pca=1}^{10} \text{pca} \)). 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 has an explained variance of 0.49 compared to the worst model, Simple PT HAKE-MAKE, with explained variance of 0.34. Coincidentally, 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

---

38We use the scikit-learn implementation [60] based on [72].

---

<table>
<thead>
<tr>
<th>Model</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>YI</th>
<th>( YI_{\text{max}} )</th>
<th>( \tau_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple FT HAKE-DistMult</td>
<td>0.947 ± 0.014</td>
<td>0.667 ± 0.02</td>
<td>0.614 ± 0.013</td>
<td>0.645 ± 0.011</td>
<td>0.736 ± 0.087</td>
</tr>
<tr>
<td>Simple FT ConvKB-DistMult</td>
<td>0.947 ± 0.012</td>
<td>0.662 ± 0.035</td>
<td>0.609 ± 0.031</td>
<td>0.634 ± 0.026</td>
<td>0.701 ± 0.132</td>
</tr>
<tr>
<td>Simple FT ConvKB-TransE</td>
<td>0.934 ± 0.009</td>
<td>0.68 ± 0.018</td>
<td>0.615 ± 0.014</td>
<td>0.642 ± 0.015</td>
<td>0.687 ± 0.065</td>
</tr>
<tr>
<td>Simple FT ConvE-RotaE</td>
<td>0.915 ± 0.013</td>
<td>0.454 ± 0.028</td>
<td>0.369 ± 0.027</td>
<td>0.402 ± 0.028</td>
<td>0.658 ± 0.083</td>
</tr>
<tr>
<td>Simple FT HAKE-MAKE</td>
<td>0.931 ± 0.009</td>
<td>0.19 ± 0.036</td>
<td>0.049 ± 0.038</td>
<td>0.171 ± 0.038</td>
<td>0.882 ± 0.127</td>
</tr>
<tr>
<td>Complex one-hot</td>
<td>0.796 ± 0.028</td>
<td>0.571 ± 0.041</td>
<td>0.367 ± 0.054</td>
<td>0.398 ± 0.043</td>
<td>0.526 ± 0.076</td>
</tr>
<tr>
<td>Complex FT HAKE-DistMult</td>
<td>0.969 ± 0.016</td>
<td>0.642 ± 0.044</td>
<td>0.61 ± 0.034</td>
<td>0.643 ± 0.026</td>
<td>0.675 ± 0.105</td>
</tr>
<tr>
<td>Complex FT pRotatE-ComplEx</td>
<td>0.929 ± 0.024</td>
<td>0.668 ± 0.048</td>
<td>0.597 ± 0.048</td>
<td>0.62 ± 0.046</td>
<td>0.526 ± 0.145</td>
</tr>
<tr>
<td>Complex FT ConvKB-DistMult</td>
<td>0.965 ± 0.013</td>
<td>0.631 ± 0.078</td>
<td>0.597 ± 0.07</td>
<td>0.627 ± 0.039</td>
<td>0.59 ± 0.149</td>
</tr>
<tr>
<td>Complex FT ComplEx-HoE</td>
<td>0.991 ± 0.01</td>
<td>0.237 ± 0.106</td>
<td>0.228 ± 0.098</td>
<td>0.45 ± 0.028</td>
<td>0.721 ± 0.047</td>
</tr>
<tr>
<td>Complex FT ComplEx-MAKE</td>
<td>0.9 ± 0.055</td>
<td>0.097 ± 0.047</td>
<td>0.003 ± 0.064</td>
<td>0.133 ± 0.081</td>
<td>0.696 ± 0.22</td>
</tr>
<tr>
<td>Complex FT HAKE-DistMult</td>
<td>0.932 ± 0.011</td>
<td>0.69 ± 0.024</td>
<td>0.622 ± 0.023</td>
<td>0.652 ± 0.022</td>
<td>0.706 ± 0.134</td>
</tr>
<tr>
<td>Complex FT pRotatE-ComplEx</td>
<td>0.931 ± 0.025</td>
<td>0.672 ± 0.042</td>
<td>0.602 ± 0.045</td>
<td>0.631 ± 0.037</td>
<td>0.627 ± 0.157</td>
</tr>
<tr>
<td>Complex FT ConvKB-DistMult</td>
<td>0.953 ± 0.008</td>
<td>0.642 ± 0.027</td>
<td>0.596 ± 0.027</td>
<td>0.625 ± 0.028</td>
<td>0.753 ± 0.138</td>
</tr>
<tr>
<td>Complex FT ComplEx-HoE</td>
<td>0.898 ± 0.035</td>
<td>0.591 ± 0.064</td>
<td>0.489 ± 0.042</td>
<td>0.521 ± 0.027</td>
<td>0.612 ± 0.156</td>
</tr>
<tr>
<td>Complex FT ComplEx-MAKE</td>
<td>0.88 ± 0.032</td>
<td>0.255 ± 0.026</td>
<td>0.135 ± 0.034</td>
<td>0.204 ± 0.06</td>
<td>0.775 ± 0.268</td>
</tr>
</tbody>
</table>

Table 13
Prediction results for sampling strategy (iii). Same notation as Table 11.
## Table 14

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

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

## Table 15

<table>
<thead>
<tr>
<th>KGE model</th>
<th># uses (i)</th>
<th># uses (ii)</th>
<th># uses (iii)</th>
<th># uses (iv)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistMult</td>
<td>1/0</td>
<td>0/1</td>
<td>1/7</td>
<td>0/4</td>
</tr>
<tr>
<td>ComplEx</td>
<td>1/1</td>
<td>1/3</td>
<td>2/1</td>
<td>1/5</td>
</tr>
<tr>
<td>HolE</td>
<td>2/0</td>
<td>1/0</td>
<td>1/0</td>
<td>1/0</td>
</tr>
<tr>
<td>TransE</td>
<td>4/1</td>
<td>2/4</td>
<td>4/8</td>
<td>2/9</td>
</tr>
<tr>
<td>RotatE</td>
<td>0/0</td>
<td>0/0</td>
<td>0/0</td>
<td>0/0</td>
</tr>
<tr>
<td>pRotatE</td>
<td>1/0</td>
<td>1/0</td>
<td>1/0</td>
<td>3/0</td>
</tr>
<tr>
<td>HAKE</td>
<td>2/8</td>
<td>3/5</td>
<td>1/0</td>
<td>2/0</td>
</tr>
<tr>
<td>Total geometric</td>
<td>4/8</td>
<td>6/5</td>
<td>3/2</td>
<td>5/0</td>
</tr>
<tr>
<td>ConvKB</td>
<td>1/1</td>
<td>0/1</td>
<td>2/0</td>
<td>0/1</td>
</tr>
<tr>
<td>ConvE</td>
<td>1/0</td>
<td>2/0</td>
<td>1/0</td>
<td>2/0</td>
</tr>
<tr>
<td>Total convolution</td>
<td>2/1</td>
<td>2/1</td>
<td>3/0</td>
<td>2/1</td>
</tr>
</tbody>
</table>

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 KGc and one for KGg, such that there is a total of 20 models per sampling strategy. Notation: ‘used in KGc’ ‘used in KGg’, e.g., HAKE, 2/8 in sampling strategy (i), indicates that HAKE is used to embed KGc 2 out of top-10 combinations and it is used to embed KGg 8 out of top-10 combinations.
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., $Y_{I_{\text{max}}}$ as seen in Figure 8, the rate of change is reduced compared to $Y_{I}$ in Figure 6.
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.*, *carfentrazone-ethyl*) to a flower (*i.e.*, *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.*, 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 transforma-
Table 16

<table>
<thead>
<tr>
<th>Chemical</th>
<th>Species</th>
<th>log((\chi))</th>
<th>Predicted</th>
<th>Lethal</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001556 (hexachlorocyclohexane)</td>
<td>59899 (walking catfish)</td>
<td>-3.4</td>
<td>0.97</td>
<td>1 (yes)</td>
<td>TP</td>
</tr>
<tr>
<td>C037925 (benothiocarb)</td>
<td>7965 (sea urchins)</td>
<td>0.9</td>
<td>0.2</td>
<td>0 (no)</td>
<td>TN</td>
</tr>
<tr>
<td>D026023 (permethrin)</td>
<td>378420 (bivalves)</td>
<td>0.7</td>
<td>0.96</td>
<td>1 (yes)</td>
<td>TP</td>
</tr>
<tr>
<td>D011189 (potassium chloride)</td>
<td>938113 (megacyclops viridis)</td>
<td>6.7</td>
<td>0.27</td>
<td>1 (yes)</td>
<td>FN</td>
</tr>
<tr>
<td>C427526 (carfentrazone-ethyl)</td>
<td>208866 (eudicots)</td>
<td>-0.9</td>
<td>0.82</td>
<td>0 (no)</td>
<td>FP</td>
</tr>
<tr>
<td>D010278 (parathion)</td>
<td>201691 (green sunfish)</td>
<td>-0.9</td>
<td>0.86</td>
<td>0 (no)</td>
<td>FP</td>
</tr>
</tbody>
</table>

Example predictions by Complex FT HolE-DistMult (best model) for sampling strategy (iv).

Table 16

<table>
<thead>
<tr>
<th>Chemical</th>
<th>Species</th>
<th>log((\chi))</th>
<th>Predicted</th>
<th>Lethal</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001556 (hexachlorocyclohexane)</td>
<td>59899 (walking catfish)</td>
<td>-3.4</td>
<td>0.97</td>
<td>1 (yes)</td>
<td>TP</td>
</tr>
<tr>
<td>C037925 (benothiocarb)</td>
<td>7965 (sea urchins)</td>
<td>0.9</td>
<td>0.2</td>
<td>0 (no)</td>
<td>TN</td>
</tr>
<tr>
<td>D026023 (permethrin)</td>
<td>378420 (bivalves)</td>
<td>0.7</td>
<td>0.96</td>
<td>1 (yes)</td>
<td>TP</td>
</tr>
<tr>
<td>D011189 (potassium chloride)</td>
<td>938113 (megacyclops viridis)</td>
<td>6.7</td>
<td>0.27</td>
<td>1 (yes)</td>
<td>FN</td>
</tr>
<tr>
<td>C427526 (carfentrazone-ethyl)</td>
<td>208866 (eudicots)</td>
<td>-0.9</td>
<td>0.82</td>
<td>0 (no)</td>
<td>FP</td>
</tr>
<tr>
<td>D010278 (parathion)</td>
<td>201691 (green sunfish)</td>
<td>-0.9</td>
<td>0.86</td>
<td>0 (no)</td>
<td>FP</td>
</tr>
</tbody>
</table>

Example predictions by Complex FT HolE-DistMult (best model) for sampling strategy (iv).

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 unfeasible 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 represents 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 5.2, 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 KGs as described in Section 5).

However, we include the full KGE and KGs.

All the material related to this project is available at https://github.com/NIV A-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 UK, Siemens AG, and the EPSRC projects ANALOG (EP/P025943/1), OASIS (EP/S032347/1), UK FIRES (EP/S019111/1) and the AIDA project (Alan Turing Institute).

References


[65] A. K. Sharma, G. N. Srivastava, A. Roy, and V. K. Sharma. Toxin: A toxicity prediction tool for small molecules de-


### 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, \( \|\cdot\|_n \) for the \( n \)-th norm, \( \langle x, y \rangle \) for the inner product (dot product) between \( x \) and \( y \), \( \|x\| \) is the concatenation of \( x \) and \( y \), \( x \) indicates the reshape of a one-dimensional vector into a two-dimensional image (not in HolE where it represent the complex conjugate), finally, \( \text{vec}(X) \) reshapes a matrix into a one-dimensional vector.

The vector representation of an entity and a relation are noted as \( e_v \) 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 [36]. 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.

\[
S_{\text{DistMult}}(sb, p, ob) = \langle e_{sb}, e_p, e_{ob} \rangle
\]
This model does not take the direction of the relation into account, that is, $SF_{\text{DistMult}}(sb, p, ob) = SF_{\text{DistMult}}(ob, p, sb)$.

**ComplEx.** This model uses 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 in DistMult is solved.

$$SF_{\text{ComplEx}}(sb, p, ob) = \langle e_{sb}, e_p, e_{ob} \rangle$$

$$= \langle \Re(e_{sb}) + i\Im(e_{sb}), \Re(e_{sb}) \rangle$$

$$+ \langle i\Im(e_{sb}), \Re(e_p) + i\Im(e_{sb}) \rangle$$

$$= \langle \Re(e_{sb}), \Re(e_p), \Re(e_{ob}) \rangle$$

$$+ \langle \Im(e_{sb}), \Re(e_p), \Im(e_{ob}) \rangle$$

$$+ \langle \Re(e_{sb}), \Im(e_p), \Re(e_{ob}) \rangle$$

$$+ \langle \Im(e_{sb}), \Im(e_p), \Re(e_{ob}) \rangle$$

(24)

where $i = \sqrt{-1}$ and, $\Re(x)$ and $\Im(x)$ are the real and complex parts of $x$, respectively. We can easily see that $SF_{\text{ComplEx}}(e_{sb}, e_p, e_{ob}) = SF_{\text{DistMult}}(e_{sb}, e_p, e_{ob})$ if $\Im(e_{sb}) = \Im(e_p) = \Im(e_{ob}) = 0$.

**HolE.** The Holographic embedding model is described in [55], and use a circuar correlation scoring function

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

(25)

$$e_{sb} \circ e_{ob} = F^{-1}[F(e_{sb}) \circ F(e_{ob})]$$

(26)

where $F$ and $F^{-1}$ are the Fourier transform and its inverse, for this model we use $\overline{x}$ as the elementwise complex conjugate, $\circ$ denotes the Hadamard product (element-wise). HolE has been shown to be equivalent to ComplEx [30], and therefore, we expect the performance to be similar.

### A.3. Geometric models

**TransE.** The translational model has the scoring function [10]

$$SF_{\text{TransE}}(sb, p, ob) = ||e_{sb} + e_p - e_{ob}||_n.$$  

(27)

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 [70]. The scoring function of RotatE is defined as

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

$$= ||\Re(e_{sb}) \cos(\theta_p) + i\sin(\theta_p) - e_{ob}||_n$$

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

$$: SF_{\text{RotatE}}(sb, p, ob) = ||\Re(e_{sb}) \sin(\theta_p) + \Im(e_{sb}) \cos(\theta_p) - \Im(e_{ob})||_n.$$  

(28)

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 [70]. pRotatE has the scoring function

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

(29)

where $\theta_x = \angle e_x$ (phase of $e_x$) 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 [86]. 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

$$SF_{\text{pRotatE}}(sb, p, ob) = |||e_{sb}| \circ |e_p| - |e_{ob}||_n$$

$$+ ||\sin(\frac{\theta_{sb} + \theta_p - \theta_{ob}}{2})||_1$$

(30)

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 [86]. 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 \( X * \omega \). 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 explicit state it. Moreover, dropout layers are used between every convolutional and dense layer.

**ConvKB.** The scoring function of ConvKB [54] use a single convolutional layer and a single dense layer

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

(31)

where \( \text{vec}(x) \) 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 [19] only perform convolution over the subject and predicate image (concatenated and reshaped) and multiples the output dense layer with the object vector as such

\[
SF_{\text{ConvE}}(sb, p, ob) = f(\text{vec}(f([e_{sb}; e_p] * \omega))(W)e_{ob}^T).
\]

(32)

where \( \mathbf{x} \) 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, [19] 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 show 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 all entities and sample negative examples from this set [21], i.e., we sample from \( \{sb, p, ob'\} \notin KG, ob' \in \mathcal{E} \). This can be expanded to the stochastic local closed world assumption, which corrupt both the subject and the object of true triples (illustrated by Fig. 3 in [4]). The number of negative samples sampled per positive sample is controlled by a hyper-parameter. However, [36] 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 \((-1 \text{ for false and } 1 \text{ for true})\) 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.

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

(34)

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

\[
L_{H_2} = \sum_{t^+ \in X^+} \sum_{t^- \in X^-} \left[\gamma + S(t^+) - S(t^-)\right]^+.
\]

(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

\[
L_{L_2} = \sum_{t^+ \in X^+} \sum_{t^- \in X^-} \log(1 + \exp(S(t^-) - S(t^+))).
\]

(36)
A.6. Implementation

We have implemented the KGE models in Keras [16] 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.