Lattice-based $\mathcal{ALC}$ ontology embeddings with saturation

Fernando Zhapa-Camacho [a] and Robert Hoehndorf [a,b,c,*]

[a] *Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), King Abdullah University of Science and Technology, 23955-6900, Thuwal, Saudi Arabia*

[b] *Center of Excellence for Smart Health, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*

[c] *Center of Excellence for Generative AI, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*

E-mails: fernando.zhapacamacho@kaust.edu.sa, robert.hoehndorf@kaust.edu.sa

Abstract. Generating vector representations (embeddings) of OWL ontologies is a growing task due to its applications in predicting missing facts and knowledge-enhanced learning in fields such as bioinformatics. The underlying semantics of OWL ontologies are expressed using Description Logics (DLs). Initial approaches to generate embeddings relied on constructing a graph out of ontologies, neglecting the semantics of the logic therein. Recent semantic-preserving embedding methods often target lightweight DL languages like $\mathcal{EL}^{++}$, ignoring more expressive information in ontologies. Although some approaches aim to embed more descriptive DLs like $\mathcal{ALC}$, those methods require the existence of individuals, while many real-world ontologies are devoid of them. We propose an ontology embedding method for the $\mathcal{ALC}$ DL language that considers the lattice structure of concept descriptions. We use connections between DL and Category Theory to materialize the lattice structure and embed it using an order-preserving embedding method. We show that our method outperforms state-of-the-art methods in several knowledge base completion tasks. This is an extended version of our previous work [74] where we incoporate saturation procedures that increase the information within the constructed lattices. We make our code and data available at https://github.com/bio-ontology-research-group/catE.

Keywords: Ontology embedding, Knowledge Base Completion, Neuro-symbolic AI

1. Introduction

Ontologies are usually developed and maintained by manual curation of experts and therefore the knowledge therein can be inconsistent or incomplete. Traditionally, symbolic reasoners are used to test for consistency of the knowledge within ontologies and to infer new statements. However, they are designed to infer statements that are logically entailed from the ontology or knowledge base; in some cases, it is useful to also suggest axioms that are probably true but not entailed, leading to the task of “ontology completion” or “knowledge base completion”.

From the viewpoint of knowledge graph completion [21], we can initially define knowledge base completion as the task of predicting “missing” or “novel” axioms in a knowledge base (or ontology). “Novel” may be understood temporally as axioms that are added at a later time to a knowledge base, or, more commonly, with respect to existing axioms in the knowledge base. However, unlike knowledge graphs, a knowledge base (ontology) has an infinitely

*Corresponding author. E-mail: robert.hoehndorf@kaust.edu.sa.

large deductive closure with deductively entailed statements. Those statements can be considered “novel” because they do not exist in the knowledge base but can effectively be generated by a deductive reasoner. Therefore, knowledge base completion can have a two-fold presentation: (1) knowledge base completion as approximate entailment, where the completion system first generates the deductively entailed statements, and then, with potentially lower confidence, the system generates the non-entailed but probable statements, and (2) the completion system generates only non-entailed statements and, optionally, has access to information to the deductive closure.

Transversally, knowledge base completion methods can be evaluated based on the type of axioms to complete. We distinguish between two sub-tasks: “TBox completion”, when the axioms to generate are terminological and are of the form $C \sqsubseteq D$, and “ABox completion”, when the axioms to generate are assertional and are of the form $C(a)$ or $r(a, b)$. TBox completion systems have been proposed as supporting tools to assist or automate ontology curation procedures [10, 19] or to match concepts between ontologies [19]. ABox completion systems are evaluated alongside neuro-symbolic reasoners in challenges like SemREC [10]. Furthermore, ABox completion can be regarded as knowledge graph completion enhanced with ontological knowledge [33].

Several neuro-symbolic approaches have been developed to perform the knowledge base completion tasks [18, 19, 37, 41], and most are based on generating embeddings that preserve some logical properties of a knowledge base. Methods which perform knowledge base completion follow different strategies. One type of methods corresponds to transforming ontology axioms into graphs. Under this approach, axioms in a DL knowledge base are transformed into a graph and then knowledge graph completion methods are applied [18]. Although this strategy has proved to be useful, this set of methods does not capture *all* information in axioms and the embedding process is usually not invertible [73]; therefore, these methods do not allow exact inference of axioms and are often used for similarity-based tasks.

Another type of methods for embedding DL knowledge bases constructs an approximate model of the knowledge base. ELEmbeddings [41] represent concepts as $n-$dimensional balls and roles are represented as geometric translations of concepts. By modifying the geometric structure from balls to boxes, methods such as BoxEL [71] guarantee intersectional closure of concepts (i.e., the intersection of two boxes is a box). However, representing roles as translations can only encode one-to-one relations. Therefore, Box^{2}EL [37] represents roles as two boxes, representing the domain and the codomain of the role, respectively. This representation enables encoding many-to-many relations. However, all these methods target the $\mathcal{EL}^{++}$ language, which is a lightweight language that does not support the construction of axioms involving full negation or universal restrictions, therefore they cannot leverage more expressive statements in DL knowledge bases. In this regard, methods such as FALCON [65], which is a method similar to Logic Tensor Networks [9], can construct an approximate model for $\mathcal{ALC}$ knowledge bases. FALCON represents concepts as fuzzy sets and treats logical connectives as fuzzy operators [66]. However, FALCON requires the existence of individuals to populate the fuzzy sets, which is a limiting factor in cases involving knowledge bases without individuals such as the Gene Ontology (GO). Another approach for modeling the $\mathcal{ALC}$ language is found in [53] with a theoretical analysis on the use of axis-aligned cones to represent ontology concepts.

1.1. Proposed approach: lattice-preserving embeddings

To overcome limitations of current ontology embedding approaches, we propose CatE, a lattice-preserving embedding method for the $\mathcal{ALC}$ language. Our approach relies on the fact that the concept descriptions in a DL knowledge base can be arranged in a lattice structure. The lattice construction of DL concepts can be formulated in the context of Formal Concept Analysis [7], using connections between DL and Modal Logic [6, 61, 68] or using connections between DL and Category Theory [16, 27]. We use the category-theoretical formulation and construct a lattice out of all concept descriptions that are sub-concepts of any concept description in the knowledge base. After materializing the lattice we represent its elements as vectors in an ordered-vector space. To enforce the ordered structure of the vector space, we use an *order-embedding method*. We apply CatE and show that it can outperform state-of-the-art methods in the different forms of the knowledge base completion task. Additionally, arranging concept descriptions in a lattice enables the application of (partial) procedures that can introduce new information to the lattice in terms of new elements and morphisms. We apply partial saturation to the lattice and show that these procedures can improve the knowledge completion performance on some metrics such as mean reciprocal rank. Our contributions are the following:

- We propose an embedding method for $\mathcal{ALC}$ knowledge bases that preserves the lattice structure of the semantics of concept descriptions.
- We show that our method can perform competitively on generating statements in the deductive closure and generating probable statements.
- We show that our method can perform competitively in both TBox and ABox completion tasks.
- We show that partial saturation procedures can enhance the embedding representation of ontology concept descriptions.

2. Preliminaries

2.1. Description Logics

A Description Logic signature $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ consists of a set of concept names $\mathbf{C}$, a set of role names $\mathbf{R}$, and a set of individual names $\mathbf{I}$. In the Description Logic $\mathcal{ALC}$, all concept names are concept descriptions; if A and B are concept descriptions, r a role name, and a, b are individual names, then $A \sqcap B$, $A \sqcup B$, $\neg A$, $\exists r.A$, and $\forall r.A$ are concept descriptions; $A \sqsubseteq B$, $A(a)$ and $r(a, b)$ are axioms. A set of axioms is an $\mathcal{ALC}$ theory [8].

An interpretation $\mathcal{I}$ of an $\mathcal{ALC}$ theory consists of an interpretation domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ such that for every concept name $C \in \mathbf{C}$, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; for every individual name $a \in \mathbf{I}$, $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; and every role name $r \in \mathbf{R}$, $r^{\mathcal{I}} \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$; and, inductively:

$$
\begin{aligned}
\bot^{\mathcal{I}} &= \emptyset \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\neg A)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}, \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
(\exists r.C)^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \mid \exists b.((a, b) \in r^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}) \right\} \\
(\forall r.C)^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \mid \forall b.((a, b) \in r^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}) \right\}
\end{aligned}
$$

An interpretation $\mathcal{I}$ is a model for an axiom $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, for an axiom $B(a)$ iff $a^{\mathcal{I}} \in B^{\mathcal{I}}$, and for an axiom $r(a, b)$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ [8]. Given an $\mathcal{ALC}$ theory T, an axiom is entailed from T if it is true in all models of T.

3. Construction of the Lattice Structure

A *preorder* $(P, \leqslant)$ contains a set P equipped with a reflexive and transitive binary relation $\leqslant$. A partial order is a preorder that is also antisymmetric. A lattice is a partially ordered set where each two-element subset has a least upper bound and greatest lower bound. If a lattice has a greatest element, it is denoted $\top$, and if it has a least element it is denoted $\bot$ [25].

In a $\mathcal{ALC}$ theory $\mathcal{T}$, the set $\mathbf{C}$ of concept names can be used to create arbitrarily complex and infinitely many concept descriptions. We consider only the concept descriptions in the knowledge base with their sub-expressions and call this set $\tilde{\mathbf{C}}$. We furthermore denote $\tilde{\mathbf{C}}^{\mathcal{I}} = \{C^{\mathcal{I}} \mid C \in \tilde{\mathbf{C}}\}$.

The pair $(\tilde{\mathbf{C}}^{\mathcal{I}}, \subseteq)$ can form a lattice where concept descriptions $C^{\mathcal{I}}, D^{\mathcal{I}} \in \tilde{\mathbf{C}}^{\mathcal{I}}$ stand in a relationship if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Within models of $\mathcal{ALC}$ theories, the relation $\subseteq$ is reflexive and transitive. For a pair of concepts descriptions $A^{\mathcal{I}}, B \in \tilde{\mathbf{C}}^{\mathcal{I}}$, the least upper bound is denoted as $(A \cup B)^{\mathcal{I}}$ and the greatest lower bound is denoted using $(A \cap B)^{\mathcal{I}}$. Additionally, for any concept description X it holds $\bot^{\mathcal{I}} \subseteq X^{\mathcal{I}} \subseteq \top^{\mathcal{I}}$.

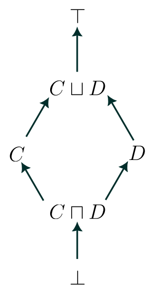


Fig. 1. Lattice representation. $\bot$ is the bottom element and $\top$ is to top element. Arrows represent the $\sqsubseteq$ operator.

To represent the lattice $(\tilde{\mathbf{C}}^{\mathcal{I}}, \subseteq)$, we use the syntactic representation of the axioms (where the operator is $\sqsubseteq$ and not $\subseteq$) and denote it as $(\tilde{\mathbf{C}}, \sqsubseteq)$ (Figure 1). The representation based on $\sqsubseteq$ does not hold all the properties of lattices; however, it is used as an intermediate structure between the lattice $(\tilde{\mathbf{C}}^{\mathcal{I}}, \subseteq)$ and the embedding space which will be introduced later (Section 3.2).

The concepts in $\tilde{\mathbf{C}}$ are materialized following a recursive process and, depending on the type of concept descriptions, $\tilde{\mathbf{C}}$ can be extended with new elements. We rely on connections between DL and Category Theory described in [27].

Intersection of concepts: Given a concept description $A \sqcap B$ in the theory, we add the following relationships to $(\tilde{\mathbf{C}}, \sqsubseteq)$: $A \sqcap B \sqsubseteq A$ and $A \sqcap B \sqsubseteq B$. Additionally, for any X, if relationships $X \sqsubseteq A \sqcap B$ are found in $(\tilde{\mathbf{C}}, \sqsubseteq)$, we add the relationships $X \sqsubseteq A$ and $X \sqsubseteq B$ (Figure 2a). Concepts A, B are processed recursively.

Union of concepts: Given a concept description $A \sqcup B$ in the theory, we add the following relationships to $(\tilde{\mathbf{C}}, \sqsubseteq)$: $A \sqsubseteq A \sqcup B$ and $B \sqsubseteq A \sqcup B$. Additionally, for any X, if relationships $A \sqcup B \sqsubseteq X$ are found in $(\tilde{\mathbf{C}}, \sqsubseteq)$, we add the relationships $A \sqsubseteq X$ and $B \sqsubseteq X$ (Figure 2b). Concepts A, B are processed recursively.

Negation of concepts: Given a concept $\neg C$, elements $C \sqcap \neg C$ and $C \sqcup \neg C$ are added to Є. The relationships $C \sqcap \neg C \sqsubseteq \bot$, $\top \sqsubseteq C \sqcup \neg C$ are added to $(\tilde{\mathbf{C}}, \sqsubseteq)$. Additionally, for any X, if the relationship $C \sqcap X \sqsubseteq \bot$ is found in $(\tilde{\mathbf{C}}, \sqsubseteq)$, we add the relationship $X \sqsubseteq \neg C$, and if the relationship $\top \sqsubseteq C \sqcup X$ is found in $(\tilde{\mathbf{C}}, \sqsubseteq)$, we add the relationship $\neg C \sqsubseteq X$ (Figure 2c). The concept C is processed recursively.

Existential restriction of concepts: First, an auxiliary preorder is constructed for DL roles, denoted as $(\tilde{\mathbf{R}}, \sqsubseteq)$. In this preorder, elements r, s stand in a relationship $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ or if $r \sqsubseteq s$ is entailed. $\tilde{\mathbf{R}}$ is extended from $\mathbf{R}$ during the lattice construction process. For any role r represented in $\tilde{\mathbf{R}}$, elements $domain(r)$ and $codomain(r)$ are added to $\tilde{\mathbf{C}}$. Given a concept description $\exists r.C$, the relationship $r_{\exists r.C} \sqsubseteq r$ is added to $(\tilde{\mathbf{R}}, \sqsubseteq)$. Relationships $codomain(r_{\exists r.C}) \sqsubseteq C$, $domain(r_{\exists r.C}) \sqsubseteq \exists r.C$ and $\exists r.C \sqsubseteq domain(r_{\exists r.C})$ are added to $(\tilde{\mathbf{C}}, \sqsubseteq)$. Additionally, if there are roles $s \in \tilde{\mathbf{R}}$ with relationships $s \sqsubseteq r$ and $codomain(r) \sqsubseteq C$, the relationship $domain(s) \sqsubseteq domain(r_{\exists r.C})$ is added to $(\tilde{\mathbf{C}}, \sqsubseteq)$. The concept C is processed recursively.

Universal restriction of concepts: Given a concept description $\forall r.C$, the element $\neg\exists r.\neg C$ is added to $\tilde{\mathbf{C}}$ with relationships $\forall r.C \sqsubseteq \neg\exists r.\neg C$ and $\neg\exists r.\neg C \sqsubseteq \forall r.C$. Furthermore, if there are roles $s \in \tilde{\mathbf{R}}$ with relationships $s \sqsubseteq r$ and $domain(s) \sqsubseteq \forall r.C$, the relationship $codomain(r) \sqsubseteq C$ is added to $(\tilde{\mathbf{C}}, \sqsubseteq)$. Concepts $\neg\exists r.\neg C$, $\neg C$ and C are processed recursively.

Subsumption axioms: Axioms $C \sqsubseteq D$ are incorporated directly to the lattice. Additionally, relationships $\top \sqsubseteq \neg C \sqcup D$ are added to $(\tilde{\mathbf{C}}, \sqsubseteq)$. Concepts C and D are processed recursively.

Class assertion axioms: Given an axiom $C(a)$, we construct the element $\{a\}$ in $\tilde{\mathbf{C}}$ with the following relationships: $\bot \sqsubseteq \{a\}$, $\{a\} \sqsubseteq C$ and $\{a\} \sqsubseteq \top$.

Role assertion axioms: Given an axiom $r(a, b)$, we construct elements $\{a\}, \{b\}$ in $\tilde{\mathbf{C}}$ with the following relationships: $\bot \sqsubseteq \{a\}$, $\{a\} \sqsubseteq \top$, $\bot \sqsubseteq \{b\}$, $\{b\} \sqsubseteq \top$ and $\{a\} \sqsubseteq \exists r.\{b\}$.

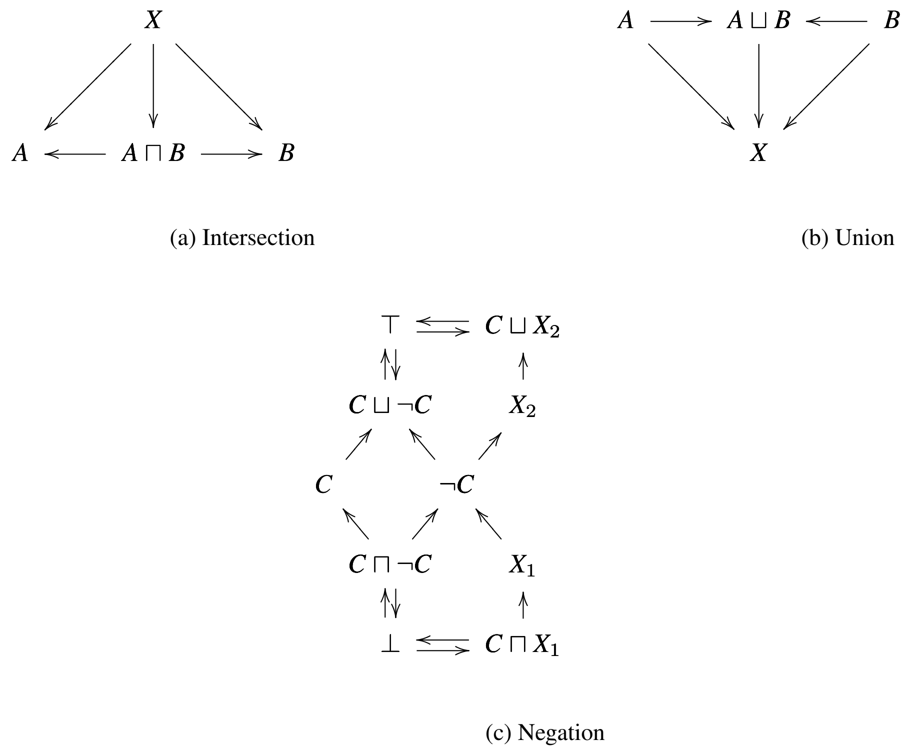


Fig. 2. Lattice representations of complex concept descriptions.

Every operator ($\sqcap$ | $\sqcup$ | $\neg$ | $\exists$ | $\forall$ |$\sqsubseteq$) introduces a constant number of elements into $\tilde{\mathbf{C}}$ and a constant number of relationships in $(\tilde{\mathbf{C}}, \sqsubseteq)$. Therefore, for a formula in the knowledge base with n operators the space and time complexity to process it is $O(n)$.

3.1. Saturation procedures

The lattice construction process is not complete in the sense that we consider a subset $\tilde{\mathbf{C}}$ from the infinite set $\mathbf{C}$ of possible concept descriptions. Additional concept descriptions that can be generated by deduction rules will not have a representation in the lattice. While extra information might improve the quality of embeddings, the time and space requirements to construct and process the lattice will be bigger. We study the impact of adding new information to the lattice by introducing saturation procedures.

By saturation we refer to the process of adding new elements and morphisms to the lattice until a fixed point is reached. However, due to practical limitations, we apply saturation rules partially and the fixed point might not be actually obtained. Since the lattice is equipped with a transitive relation, an immediate saturation rule is to compute the transitive closure of the lattice. Additionally, as specified in [16], certain deduction rules can be transformed into partial saturation procedures. We specify the rules below in the form of *precondition* $\Rightarrow$ *consequence*, where *precondition* denotes the set of morphisms existing in the lattice and *consequence* denotes the set of elements and morphisms to be added to the lattice. Therefore, for elements $C, D, E \in \tilde{\mathbf{C}}$ and for elements in $r \in \tilde{\mathbf{R}}$:

$$C \sqsubseteq \neg D \Rightarrow D \sqsubseteq \neg C \tag{1}$$

$$C \sqcap D \sqsubseteq \bot \Rightarrow C \sqsubseteq \neg D \tag{2}$$

$$\bot \sqsubseteq C \sqcup D \Rightarrow \neg C \sqsubseteq D \tag{3}$$

$$C \sqsubseteq D, D \sqsubseteq E \Rightarrow C \sqsubseteq E \tag{4}$$

$$C \sqsubseteq D \Rightarrow \exists r.C \sqsubseteq \exists r.D \tag{5}$$

Equations 1, 2, 3 are applicable to a subset of the asserted morphisms in the lattice and all of them introduce one new element to the lattice. Equation 4 corresponds to the transitive closure of the lattice and only introduces new morphisms but not new elements. Equation 5 is applicable to all morphisms in the lattice and introduces $2 \times |\tilde{\mathbf{R}}|$ elements per morphism in the lattice. Due to the large space complexity required when implementing Equation 5, we do not consider it in our analysis.

3.2. Embedding into an ordered-vector space

With the structure $(\tilde{\mathbf{C}}, \sqsubseteq)$ in place, we proceed to embed it into an ordered-vector space. This step is crucial for preserving the hierarchical relationships within the lattice, ensuring that our embeddings reflect the inherent ordering of concepts descriptions. We use an ordered-vector space $(X, \preceq)$ over $\mathbb{R}^n$ where, for elements in $a, b \in X$ with $a = (a_1, ..., a_n)$ and $b = (b_1, ..., b_n)$, $a \preceq b$ if and only if $a_1 \leqslant b_1, ..., a_n \leqslant b_n$. We show in Appendix A that $(X, \preceq)$ is an ordered-vector space.

Consequently, we introduce a parameterized function f_θ which maps objects in $(\tilde{\mathbf{C}}, \sqsubseteq)$ to the ordered-vector space $(X, \preceq)$ over $\mathbb{R}^n$. In this way, we intend f_θ to be a lattice-preserving function of $(\tilde{\mathbf{C}}, \sqsubseteq)$. Since f_θ is unknown, our task is to find the set of parameters $\theta \in \Theta$ that accommodates to the intended structure of the space X. We optimize f_θ using gradient descent. We use the following order-preserving scoring function [67]:

$$s(A, B) = ||\max(0, f_\theta(A) - f_\theta(B))||_2 \tag{6}$$

for elements $A, B \in \tilde{\mathbf{C}}$ with a relationship $A \sqsubseteq B$. If $f_\theta(A) \preceq f_\theta(B)$, then $s(A, B) = 0$, and otherwise $s(A, B) > 0$. We apply the following loss function to all relationships $A \sqsubseteq B \in (\tilde{\mathbf{C}}, \sqsubseteq)$:

$$\mathcal{L} = \sum_{A \sqsubseteq B \in (\tilde{\mathbf{C}}, \sqsubseteq)} \sum_{A \sqsubseteq B' \notin (\tilde{\mathbf{C}}, \sqsubseteq)} s(A, B) + \max(0, \gamma - s(A, B')) \tag{7}$$

Relationships $A \sqsubseteq B' \notin (\tilde{\mathbf{C}}, \sqsubseteq)$ are called negative samples and are generated by replacing B in an existing relationship $A \sqsubseteq B$ by a corrupted entity B' obtained by random sampling in a uniform distribution. The parameter γ is a margin parameter enforcing a minimum score value of the negative samples.

We show that the space X gets a partial order structure whenever the loss function $\mathcal{L} = 0$.

Theorem 1 (Lattice-preserving embeddings)**.** *Let $\mathcal{O}$ be a $\mathcal{ALC}$ theory with signature $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ and $(\tilde{\mathbf{C}}, \sqsubseteq)$ the lattice of concepts descriptions generated from $\mathcal{O}$. Let $(X, \preceq)$ be an ordered-vector space where for elements $a, b \in X$ with $a = (a_1, ..., a_n)$ and $b = (b_1, ..., b_n)$, $a \preceq b$ if and only if $a_1 \leqslant b_1, ..., a_n \leqslant b_n$. Let f_θ be a function mapping objects from $\tilde{\mathbf{C}}$ to X. If $\mathcal{L} = 0$, then f_θ is a lattice preserving function of $(\tilde{\mathbf{C}}, \sqsubseteq)$ into $(X, \preceq)$.*

Proof. Let us assume that $\mathcal{L} = 0$ and there exist a relationship $A \sqsubseteq B$ in the lattice such that $f_\theta(A) \npreceq f_\theta(B)$, meaning that the order is not preserved in the vector space X. Reordering the definition of L in Equation 7, we have that $\mathcal{L} = s(A, B) + K$, where K is a non-negative number. Therefore, since $\mathcal{L} = 0$, it follows that $s(A, B) = ||\max 0, f_\theta(A) - f_\theta(B)|| = 0$. Consequently, we have that $f_\theta(A) \preceq f_\theta(B)$, which leads to a contradiction.

Now that we have shown that any relationship $A \sqsubseteq B$ in the lattice $(\tilde{\mathbf{C}}, \sqsubseteq)$ is preserved as $f_\theta(A) \preceq f_\theta(B)$ in $(X, \preceq)$, we now verify that f_θ preserves partial-order properties:

1. Reflexivity: Let $A \sqsubseteq A$ be a relationship in $(\tilde{\mathbf{C}}, \sqsubseteq)$. Since $\mathcal{L} = 0$, it implies that $f_\theta(A) \preceq f_\theta(A)$.
2. Transitivity: Let $A \sqsubseteq B$ and $B \sqsubseteq C$ be relationships in $(\tilde{\mathbf{C}}, \sqsubseteq)$. Since $\mathcal{L} = 0$, it follows that $f_\theta(A) \preceq f_\theta(B)$ and $f_\theta(B) \preceq f_\theta(C)$ and, by the transitive property of $\preceq$, $f_\theta(A) \preceq f_\theta(C)$.

Table 1
Number of axioms in training, validation and testing ontologies, number of relationships in the corresponding training lattices and DL expressivity.

Dataset	Training	Validation	Testing	Lattice	Expressivity
ORE1	61245	7578	15157	364849	$\mathcal{EL}^{++}$
FoodOn	34224	2977	5957	631423	$\mathcal{ALC}$
GO	81844	7260	14521	1257443	$\mathcal{EL}^{++}$
PPI	351435	12038	12040	4479085	$\mathcal{EL}^{++}$

3. Antisymmetry: Let $A \sqsubseteq B$ and $B \sqsubseteq A$ be relationships in $(\tilde{\mathbf{C}}, \sqsubseteq)$. Since $\mathcal{L} = 0$, it follows that $f_\theta(A) \preceq f_\theta(B)$ and $f_\theta(B) \preceq f_\theta(A)$ and, by the antisymmetry property of $\preceq$, $f_\theta(A) = f_\theta(B)$.

□

The embedding f_θ preserves a lattice structure. The lattice CatE preserves is generated from the objects and morphisms of the category that provides the semantics for a theory T. This category is shown to be compatible with classical semantics in the sense that the theory T is category-theoretically unsatisfiable if and only if T is set-theoretically unsatisfiable [27]. Therefore, by preserving the lattice structure, f_θ also preserves the categorical semantics, and therefore also classical semantics.

Example in $\mathbb{R}^2$*:* Consider the theories $T = \{A \sqsubseteq B, B \sqsubseteq C, C \sqcup D \sqsubseteq E\}$ and $T' = T \cup \{B \sqsubseteq \bot\}$ and their lattices shown in Figure 3a and Figure 3b, respectively. Figure 3c shows an example of embeddings in $\mathbb{R}^2$. In theory T', the concept B is unsatisfiable ($B \sqsubseteq \bot$), which implies that A is also unsatisfiable by the transitivity rule (Equation 4). This implies that, set-theoretically $A^\mathcal{I} \equiv B^\mathcal{I} \equiv \bot^\mathcal{I} \equiv \emptyset$ for all models, and category-theoretically the morphisms $A \leftrightarrow B$, $A \leftrightarrow \bot$ and $B \leftrightarrow \bot$ hold. By the antisymmetry property of f_θ, the only way to preserve the lattice structure is to make $A, B, \bot$ equivalent, which is reflected in Figure 3c where, in the embedding space of theory T', the embeddings for concepts $A, B, \bot$ get closer and, based on Equation 7, will eventually converge.

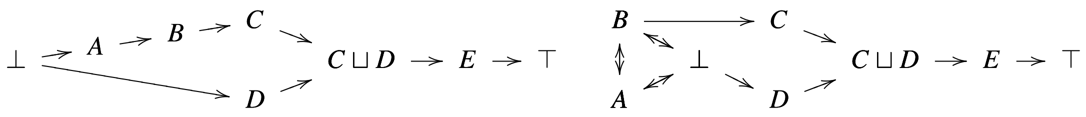
4. Evaluation

To show the effectiveness of our method, we evaluate on the following tasks: (1) generation of entailed axioms and (2) generation of probable axioms. In the task of generating entailed axioms, we use the ORE1 dataset from SemREC [10] and generate axioms of the form $C(a)$, where C is a concept name and a is an individual. In the case of generating probable axioms, we constructed datasets using GO [4] and FoodOn [26] to generate axioms of the form $C \sqsubseteq D$, where C, D are concept names. For each case, we also show that partially saturating the constructed lattice impacts the performance of axiom generation. Additionally, we applied our method to the biomedical task of predicting protein–protein interactions. This task is a form of generation of probable statements of the form $r(a, b)$, where r is a role and a, b are individuals. We show information about datasets in Table 1.

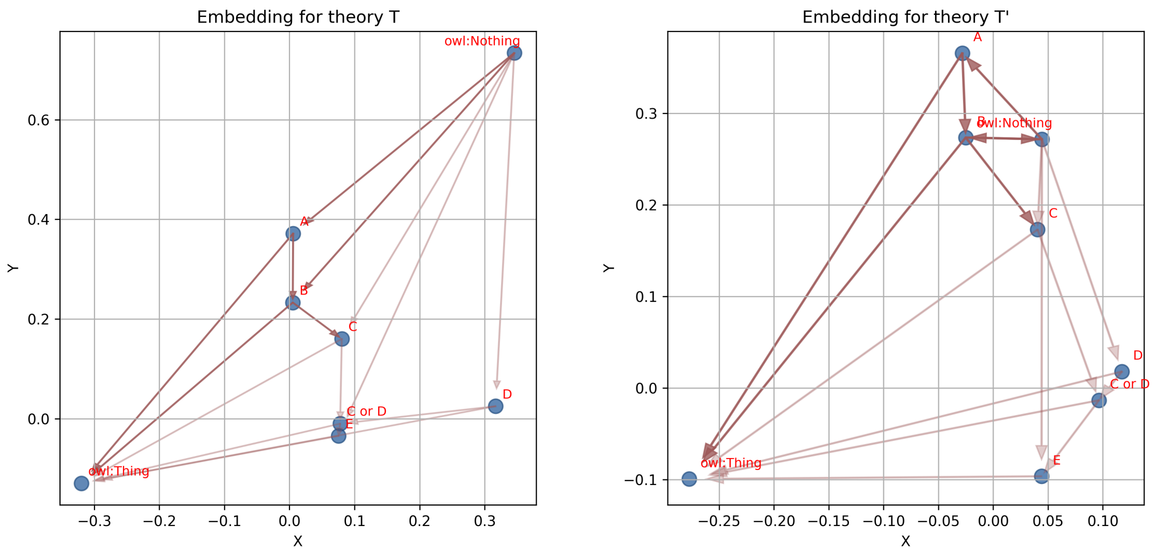
4.1. Experimental Setup

To find the optimal hyperparameters for our method, we performed a grid search over parameters: embedding dimension $\in [50, 100, 200]$, margin (γ) $\in [0, 0.01, 0.1, 1]$, number of negative samples $\in [1, 2, 4]$, batch size $\in [8192, 16384, 32768]$, and learning rate $\in [1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}]$. We used the Adam optimizer [39] with a Cyclic Learning Rate scheduler [64].

As baseline methods we selected those approaches that use only the ontology axioms, without any external knowledge such as text[18, 19]. Therefore, we selected ELEmbeddings [41] and Box^{2}EL [37] and used the implementations provided in the mOWL library [75]. Since both methods can handle only axioms in $\mathcal{EL}^{++}$, we normalized the ontologies to $\mathcal{EL}^{++}$ normal forms. In the case of FoodOn we applied our method to both the normalized $\mathcal{EL}^{++}$ and $\mathcal{ALC}$ versions. To obtain optimal parameters for baseline methods, we performed a grid search



(a) Lattice for theory T (b) Lattice for theory T'



(c) Example of embeddings in $\mathbb{R}^2$ of lattices generated by theories T (left) and T' (right)

Fig. 3. (a) and (b) represent the lattices generated from theories T and T', respectively. (c) shows the generated embeddings for objects in both theories. Notice that in the embeddings for theory T', entities $A, B, \bot$ accommodate close to each other since they become semantically equivalent.

over embedding dimension $\in [50, 100, 200]$, margin $\in [0, 0.01, 0.1]$ batch size $\in [5000, 10000, 20000]$ and learning rate $\in [1e^{-5}, 1e^{-4}, 1e^{-3}]$. Additionally, we compared with FALCON [65]; however, due to high memory and time requirements, we were unable to test different hyperparameters for this method. All selected hyperparameters are provided in the Appendix B.

We report a variety of rank-based metrics such as Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits@3, Hits@10, Hits@100 and ROC AUC.

In all tasks we report filtered metrics only and filter statements from the training set. In the task of generating axioms $C(a)$, we additionally filter statements from the deductive closure of the training set.

4.2. Generating Entailed Axioms $C(a)$

The SemREC challenge [10], which evaluates neuro-symbolic reasoners, provides a number of benchmark datasets. We selected a representative data set called ORE1. We used the ORE1 dataset to test our method on the task of predicting axioms $C(a)$, where C is a concept description and a is an individual. We perform a ranking-based evaluation, where we rank every testing statement $C(a)$ against every $C'(a)$ where C' is a named concept. We show results in Table 2, where we can see CatE performs better than baseline methods across all metrics.

Table 2

Prediction of axioms $C(a)$ where C is a concept and a is an individual. We selected the ORE1 dataset proposed in [10].

Method	MR	MRR	Hits@3	Hits@10	Hits@100	AUC
ELEmbeddings	105	0.12	0.08	0.22	0.87	0.99
Box^{2}EL	122	0.10	0.08	0.18	0.70	0.98
FALCON	603	0.02	0.00	0.02	0.34	0.92
CatE	**37**	**0.18**	**0.10**	**0.51**	**0.96**	**0.99**

Table 3

TBox completion task over axioms $C \sqsubseteq D$ in GO and FoodOn.

Dataset	Method	MR	H@10	H@100	AUC
GO	ELEmbeddings	3562	0.19	0.37	0.92
	Box^{2}EL	6621	0.01	0.07	0.85
	FALCON (5 models)	8982	0.02	0.08	0.79
	CatE	**2968**	**0.22**	**0.58**	**0.93**
FoodOn	ELEmbeddings	3336	0.25	0.38	0.88
	Box^{2}EL	2763	0.06	0.19	0.90
	FALCON (5 models)	3815	0.02	0.12	0.86
	CatE-EL	**2633**	0.29	0.43	**0.91**
	CatE	2764	**0.30**	**0.47**	0.90

4.3. *Generating Probable Axioms $C \sqsubseteq D$*

To evaluate on the task of generating probable axioms, we generate two benchmark sets following procedures designed in previous methods [18, 50]. We create two datasets using the Gene Ontology [4] and the Food Ontology [26]. In each ontology we remove 30% of the axioms $C \sqsubseteq D$ uniformly at random and distribute 10% for validation and 20% for testing. The training set contains the 70% of the subsumption axioms together with the other axioms existing in the ontology.

We focus on the prediction of subsumption axioms $C \sqsubseteq D$ and perform a rank-based evaluation ranking scores of axioms of interest $C \sqsubseteq D$ over all axioms $C \sqsubseteq D'$ where D' are named concepts. Table 3 shows the results. We can see that CatE consistently outperforms baselines in all metrics. In the case of FoodOn, we apply CatE to the $\mathcal{EL}^{++}$ (CatE-EL) and the $\mathcal{ALC}$ (CatE) versions of the ontology. CatE-EL outperforms the baselines, demonstrating that our method generate better embeddings for this specific task. Adding extra information present in the $\mathcal{ALC}$ version of FoodOn improves the Hits@k metrics.

4.4. *Protein–Protein Interaction Prediction*

Protein-protein interactions (PPIs) are direct or indirect interactions between proteins, and information about PPIs is useful in systems biology and network-based bioinformatics methods. While PPIs can be investigated experimentally, several strategies have been developed to predict them using a variety of information, including the predicted or experimentally determined functions of proteins. The functions of proteins can be represented using the GO, and if X is a class from the GO, the axiom $p_1 \sqsubseteq \exists hasFunction.X$ asserts that the class of proteins p_1 has function X. PPIs can be encoded in axioms $interacts(p_1, p_2)$ where p_1, p_2 are proteins. In order to apply our method, we need to ensure that elements $\exists interacts.p_i$ exists in the lattice for any class of proteins p_i. Therefore, we added the relationships $\bot \sqsubseteq \exists interacts.p_i$ and $\exists interacts.p_i \sqsubseteq \top$ to the lattice structure for all classes of proteins p_i. We used the PPI dataset provided in [75]. We compare our method against state-of-the-art methods such as ELEmbeddings and Box^{2}EL. [43], [71], We show the results in Table 4, where we can see that CatE is not able to outperform over baselines. The PPI benchmark relies on the assumption that the information GO acts as background knowledge to

Table 4

Protein-protein interaction prediction on Yeast. Left-side shows the results on PPI axioms. Right side shows the results on axioms $C \sqsubseteq D$ that are learned during training.

Method	PPI axioms $r(a,b)$						Axioms $C \sqsubseteq D$		
	MR	MRR	H@3	H@10	H@100	AUC	MR	H@100	AUC
ELEmbeddings	289	<u>0.10</u>	<u>0.09</u>	<u>0.25</u>	<u>0.73</u>	0.95	23812	<u>0.00</u>	0.53
Box^{2}EL	**188**	**0.17**	**0.19**	**0.43**	**0.81**	**0.97**	<u>23234</u>	<u>0.00</u>	<u>0.54</u>
CatE	<u>223</u>	0.08	0.07	0.18	0.69	<u>0.96</u>	**8936**	**0.28**	**0.82**

predict protein–protein interactions. To further investigate on this task, we evaluate how well the methods capture the hierarchy of GO functions, which are axioms of the type $C \sqsubseteq D$. We compute the deductive closure of axioms $C \sqsubseteq D$ using the ELK reasoner [38], and evaluate the capability of each method to generate the axioms in this new set. We find that ELEmbeddings and Box^{2}EL do not capture the semantics of GO axioms at all, yet they can perform PPI predictions. Originally, ELEmbeddings and Box^{2}EL are trained with negative samples just for PPI axioms, which can cause the other axioms types to converge to a trivial solution. Since CatE uses negative samples for all relationships in the lattice, it can predict PPIs while capturing other type of information in GO. Our analysis shows that predicting PPIs on its own is not sufficient to show that a particular embedding method is utilizing the background knowledge. Further analysis on embedding methods should be required, which is out of the scope of this work.

4.5. *Effect of partial saturation procedures*

To analyze the impact of the saturation procedures, we extend the lattices of the ORE1, GO and FoodOn use cases. We first experiment with the ORE1 lattice as it is the smallest one and apply three types of saturation: (a) S1, which consists of applying Equations 1, 2, 3, (b) Tr, which consists on computing the transitive closure of the lattice, and (c) S1-Tr, which consists on performing S1 followed by Tr. For GO and FoodOn use cases, which produce larger lattices, we only apply S1 because the other settings introduce a large number of elements and morphisms which make the optmization costly and also hinders the hyperparameter search. We show performance results in Table 5 and notice that the S1 procedure contributes to improve the Mean Reciprocal Rank and Hits@3 metrics in the three use cases. Additionally, for ORE1, the Tr procedure improves metrics such as Mean Rank and Hits@100; however, the combination of S1-Tr does not contribute to improve the performance.

Table 5

Impact of the application saturation procedures on the performance of generation of axioms.

Dataset	Method	MRR	H@3	H@10	H@100	MR
ORE1	CatE	<u>0.175</u>	0.097	**0.505**	<u>0.958</u>	37
	CatE-S1	**0.176**	**0.115**	<u>0.426</u>	0.884	46
	CatE-Tr	0.164	<u>0.104</u>	0.381	**0.991**	**23**
	CatE-S1-Tr	0.155	0.098	0.367	0.931	<u>30</u>
GO	CatE	<u>0.062</u>	<u>0.008</u>	<u>0.216</u>	<u>0.578</u>	**2968**
	CatE-S1	**0.066**	**0.011**	**0.226**	**0.595**	<u>3002</u>
FoodOn	CatE	<u>0.087</u>	<u>0.023</u>	**0.298**	**0.473**	**2764**
	CatE-S1	**0.094**	**0.121**	<u>0.238</u>	<u>0.419</u>	<u>3310</u>

4.6. *Effect of hyperparameters*

The time and space complexity of CatE increases linearly with the number of operators. However, the number of operators can be arbitrarily large for axioms in $\mathcal{ALC}$. Furthermore, hyperparameters such as embedding size and number of negative samples can have an impact on training and/or inference time as well as on memory consumption. In Table 4, we analyze how these hyperparameters impact on performance. We chose Hits@100 and ROC AUC metrics and show that while the embedding dimension has a direct impact performance (the higher the dimension the better the performance), the number of negative samples does not have large effect (either positive or negative).

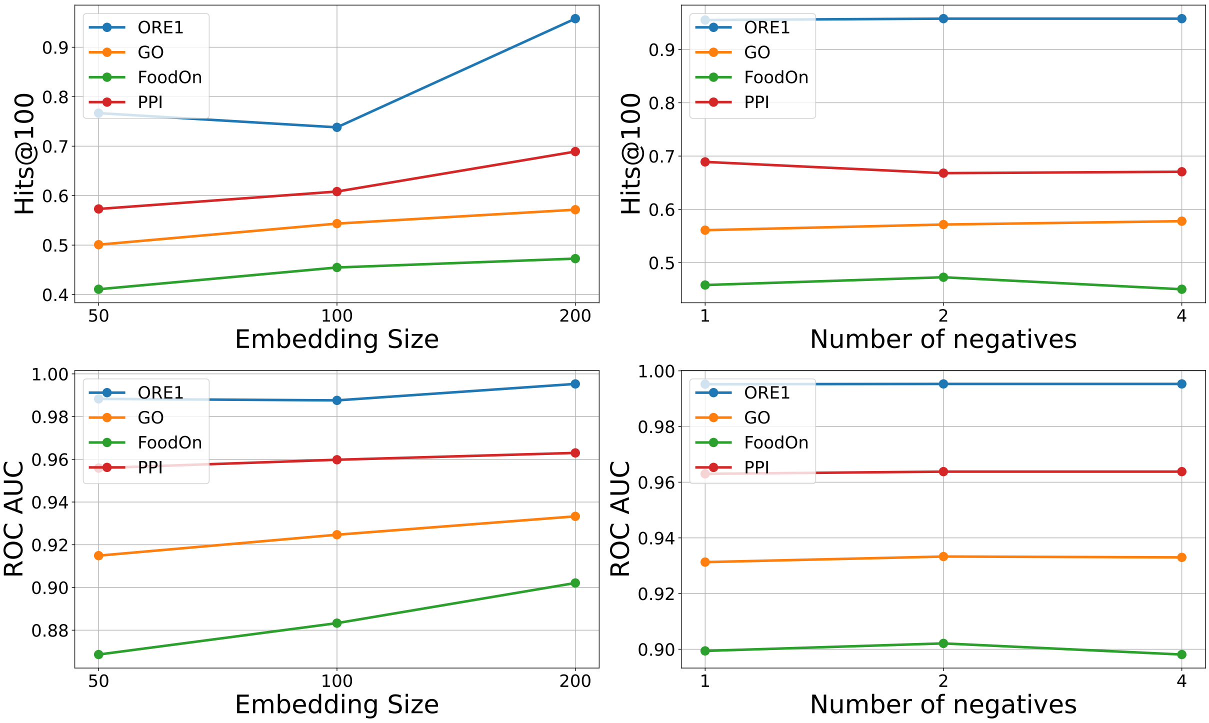


Fig. 4. Impact of embedding size and number of negatives on the Hits@100 and ROC AUC over different datasets.

5. Discussion

We have developed a method named CatE that generates embeddings for the $\mathcal{ALC}$ language. CatE consists on materializing the lattice structure of concept descriptions found in a $\mathcal{ALC}$ knowledge base. Furthermore, we use an order-preserving loss function to optimize the embedding space, and we show that when our loss function is minimized, the embedding space preserves partial order properties. We have applied our method to different forms of knowledge base completion tasks, and we showed that our method can outperform several state-of-the-art methods.

Additionally, we implemented saturation procedures to extend the lattices and the information therein. We showed that saturated versions of the lattices can improve on some metrics. However, not all the saturation rules can be applied if the knowledge bases are large because the size of the resulting lattice and the number of morphisms can hinder the application of the optimization process. A potential direction for future work can be to generating some concepts directly in the embedding space rather than explicitly materializing them within the lattice.

Current graph-based methods to embed DL knowledge bases (ontologies) construct graphs relying on syntactic information therein and the embedding process is not guaranteed to be invertible. On the other hand, methods such as ELEmbeddings, Box^{2}EL and FALCON are able to generate approximate models for DL knowledge bases. We state that CatE stands in a midpoint between both types of methods. CatE looks into the syntactical information in the knowledge base to construct a lattice and, consequently, an embedding space that is consistent to the semantics.

However, as in graph-based embeddings, CatE cannot make inferences over concepts that are not explicitly stated in the lattice. This is a limitation that was exposed in the protein–protein interaction task, where we had to add concept descriptions a priori in order to be able to make inferences over them. To mitigate this issue, future work can focus on solutions based on inductive learning over knowledge graphs, which can be applicable in the context of lattices.

6. Conclusion

We developed an embedding method for the $\mathcal{ALC}$ that preserves the lattice structure of concept descriptions. Our method materializes the lattice structure following connections between Description Logics and Category Theory. The lattice in place is embedded into an ordered-vector space. We provide empirical results that our method can perform effectively across different tasks involving knowledge base completion.

7. Acknowledgement

We acknowledge support from the KAUST Supercomputing Laboratory.

8. Funding

This work has been supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/4355-01-01, URF/1/4675-01-01, URF/1/4697-01-01, URF/1/5041-01-01, and REI/1/5334-01-01. This work was supported by the SDAIA–KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA–KAUST AI), by funding from King Abdullah University of Science and Technology (KAUST) – KAUST Center of Excellence for Smart Health (KCSH) under award number 5932, and by funding from King Abdullah University of Science and Technology (KAUST) – KAUST Center of Excellence for Generative AI under award number 5940.

References

[1] H. Abelson, G.J. Sussman and J. Sussman, *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, Massachusetts, 1985.

[2] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp and J. Lehmann, PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings, *Journal of Machine Learning Research* **22**(82) (2021), 1–6. http://jmlr.org/papers/v22/20-825.html.

[3] G. Antoniou and F. van Harmelen, *Web Ontology Language: OWL*, in: *Handbook on Ontologies*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 67–92–. ISBN 978-3-540-24750-0. doi:10.1007/978-3-540-24750-0_4.

[4] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin and G. Sherlock, Gene Ontology: tool for the unification of biology, *Nature Genetics* **25**(1) (2000), 25–29. doi:10.1038/75556.

[5] S. Awodey, *Category Theory*, 2nd edn, Oxford Logic Guides, Oxford University Press, London, England, 2010.

[6] F. Baader and C. Lutz, 13 Description logic, in: *Handbook of Modal Logic*, P. Blackburn, J. Van Benthem and F. Wolter, eds, Studies in Logic and Practical Reasoning, Vol. 3, Elsevier, 2007, pp. 757–819. ISSN 1570-2464. doi:https://doi.org/10.1016/S1570-2464(07)80016-4. https://www.sciencedirect.com/science/article/pii/S1570246407800164.

[7] F. Baader and B. Sertkaya, *Applying Formal Concept Analysis to Description Logics*, in: *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004, pp. 261–286–. ISSN 1611-3349. ISBN 9783540246510. doi:10.1007/978-3-540-24651-0_24.

[8] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P.F. Patel-Schneider (eds), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2003.

[9] S. Badreddine, A. d'Avila Garcez, L. Serafini and M. Spranger, Logic Tensor Networks, *Artificial Intelligence* **303** (2022), 103649. doi:https://doi.org/10.1016/j.artint.2021.103649. https://www.sciencedirect.com/science/article/pii/S0004370221002009.

[10] D. Banerjee, R. Usbeck, N. Mihindukulasooriya, G. Singh, R. Mutharaju and P. Kapanipathi (eds), Joint Proceedings of Scholarly QALD 2023 and SemREC 2023 co-located with 22nd International Semantic Web Conference ISWC 2023, Athens, Greece, November 6-10, 2023, in *CEUR Workshop Proceedings*, Vol. 3592, CEUR-WS.org, 2023. https://ceur-ws.org/Vol-3592.

[11] R. Baumgartner, G. Gottlob and S. Flesca, Visual Information Extraction with Lixto, in: *Proceedings of the 27th International Conference on Very Large Databases*, Morgan Kaufmann, Rome, Italy, 2001, pp. 119–128.
[12] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Advances in Neural Information Processing Systems*, Vol. 26, Curran Associates, Inc., 2013.
[13] S. Borgwardt and R. Peñaloza, Description Logics over Lattices with Multi-valued Ontologies, in: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*, T. Walsh, ed., AAAI Press, Barcelona, Spain, 2011, pp. 768–773.
[14] S. Borgwardt and R. Peñaloza, The Complexity of Lattice-Based Fuzzy Description Logics, *Journal on Data Semantics* **2**(1) (2012), 1–19. doi:10.1007/s13740-012-0013-x.
[15] R.J. Brachman and J.G. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive Science* **9**(2) (1985), 171–216.
[16] L. Brieulle, C.L. Duc and P. Vaillant, Reasoning in the Description Logic ALC under Category Semantics (Extended Abstract), in: *Proceedings of the 35th International Workshop on Description Logics (DL 2022) co-located with Federated Logic Conference (FLoC 2022), Haifa, Israel, August 7th to 10th, 2022*, O. Arieli, M. Homola, J.C. Jung and M. Mugnier, eds, CEUR Workshop Proceedings, Vol. 3263, CEUR-WS.org, 2022. https://ceur-ws.org/Vol-3263/abstract-7.pdf.
[17] J. Chen, Y. Geng, Z. Chen, I. Horrocks, J.Z. Pan and H. Chen, Knowledge-aware Zero-Shot Learning: Survey and Perspective, in: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Z. Zhou, ed., ijcai.org, 2021, pp. 4366–4373. doi:10.24963/ijcai.2021/597.
[18] J. Chen, P. Hu, E. Jimenez-Ruiz, O.M. Holter, D. Antonyrajah and I. Horrocks, OWL2Vec*: embedding of OWL ontologies, *Machine Learning* (2021). doi:10.1007/s10994-021-05997-6.
[19] J. Chen, Y. He, Y. Geng, E. Jiménez-Ruiz, H. Dong and I. Horrocks, Contextual semantic embeddings for ontology subsumption prediction, *World Wide Web* **26**(5) (2023), 2569–2591–. doi:10.1007/s11280-023-01169-9.
[20] J. Chen, A. Althagafi and R. Hoehndorf, Predicting candidate genes from phenotypes, functions and anatomical site of expression, *Bioinformatics* **37**(6) (2020), 853–860. doi:10.1093/bioinformatics/btaa879.
[21] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao and Z. Duan, Knowledge Graph Completion: A Review, *IEEE Access* **8** (2020), 192435–192456–. doi:10.1109/access.2020.3030076. http://dx.doi.org/10.1109/ACCESS.2020.3030076.
[22] F.G. Cozman and H.N. Munhoz, Some thoughts on knowledge-enhanced machine learning, *International Journal of Approximate Reasoning* **136** (2021), 308–324–. doi:10.1016/j.ijar.2021.06.003.
[23] M. Crouse, I. Abdelaziz, C. Cornelio, V. Thost, L. Wu, K. Forbus and A. Fokoue, Improving Graph Neural Network Representations of Logical Formulae with Subgraph Pooling, 2020.
[24] B.A. Davey and H.A. Priestley, *Lattices and complete lattices*, in: *Introduction to Lattices and Order*, 2nd edn, Cambridge University Press, 2002, pp. 33–64–. doi:10.1017/CBO9780511809088.004.
[25] B.A. Davey and H.A. Priestley, *Ordered sets*, in: *Introduction to Lattices and Order*, 2nd edn, Cambridge University Press, 2002, pp. 1–32–. doi:10.1017/CBO9780511809088.003.
[26] D.M. Dooley, E.J. Griffiths, G.S. Gosal, P.L. Buttigieg, R. Hoehndorf, M.C. Lange, L.M. Schriml, F.S.L. Brinkman and W.W.L. Hsiao, FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration, *npj Science of Food* **2**(1) (2018). doi:10.1038/s41538-018-0032-6.
[27] C.L. Duc, Category-theoretical Semantics of the Description Logic ALC, in: *Proceedings of the 34th International Workshop on Description Logics (DL 2021) part of Bratislava Knowledge September (BAKS 2021), Bratislava, Slovakia, September 19th to 22nd, 2021*, M. Homola, V. Ryzhikov and R.A. Schmidt, eds, CEUR Workshop Proceedings, Vol. 2954, CEUR-WS.org, 2021. https://ceur-ws.org/Vol-2954/paper-22.pdf.
[28] B. Glimm, I. Horrocks, B. Motik, G. Stoilos and Z. Wang, HermiT: An OWL 2 Reasoner, *J. Autom. Reason.* **53**(3) (2014), 245–269–. doi:10.1007/s10817-014-9305-1.
[29] V. Goranko and M. Otto, 5 Model theory of modal logic, in: *Handbook of Modal Logic*, P. Blackburn, J. Van Benthem and F. Wolter, eds, Studies in Logic and Practical Reasoning, Vol. 3, Elsevier, 2007, pp. 249–329. ISSN 1570-2464. doi:https://doi.org/10.1016/S1570-2464(07)80008-5. https://www.sciencedirect.com/science/article/pii/S1570246407800085.
[30] G. Gottlob, Complexity results for nonmonotonic logics, *Journal of Logic and Computation* **2**(3) (1992), 397–425.
[31] G. Gottlob, N. Leone and F. Scarcello, Hypertree Decompositions and Tractable Queries, *Journal of Computer and System Sciences* **64**(3) (2002), 579–627.
[32] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, *International Journal of Human-Computer Studies* **43**(5) (1995), 907–928. doi:https://doi.org/10.1006/ijhc.1995.1081. https://www.sciencedirect.com/science/article/pii/S1071581985710816.
[33] J. Hao, M. Chen, W. Yu, Y. Sun and W. Wang, Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining*, KDD '19, ACM, 2019. doi:10.1145/3292500.3330838.
[34] T. Hinnerichs and R. Hoehndorf, DTI-Voodoo: machine learning over interaction networks and ontology-based background knowledge predicts drug–target interactions, *Bioinformatics* **37**(24) (2021), 4835–4843. doi:10.1093/bioinformatics/btab548.
[35] X. Huang, J. Zhang, D. Li and P. Li, Knowledge Graph Embedding Based Question Answering, in: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 105–113–. ISBN 9781450359405. doi:10.1145/3289600.3290956.
[36] IJCAI Proceedings, IJCAI Camera Ready Submission.
[37] M. Jackermeier, J. Chen and I. Horrocks, Box^{2}EL: Concept and Role Box Embeddings for the Description Logic EL++, 2023.

[38] Y. Kazakov, M. Krötzsch and F. Simančík, The Incredible ELK, *Journal of Automated Reasoning* **53**(1) (2013), 1–61–. doi:10.1007/s10817-013-9296-3.

[39] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015.

[40] M. Kulmanov and R. Hoehndorf, DeepGOZero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms, *Bioinformatics* **38**(Supplement_1) (2022), i238–i245. doi:10.1093/bioinformatics/btac256.

[41] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric construction of models for the Description Logic EL ++, in: *International Joint Conference on Artificial Intelligence*, 2019.

[42] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric construction of models for the Description Logic EL ++, in: *International Joint Conference on Artificial Intelligence*, 2019.

[43] M. Kulmanov, F.Z. Smaili, X. Gao and R. Hoehndorf, Semantic similarity and machine learning with ontologies, *Briefings in Bioinformatics* **22**(4) (2020), bbaa199. doi:10.1093/bib/bbaa199.

[44] M. Kulmanov, F.J. Guzmán-Vega, P. Duek Roggli, L. Lane, S.T. Arold and R. Hoehndorf, Protein function prediction as approximate semantic entailment, *Nature Machine Intelligence* **6**(2) (2024), 220–228–. doi:10.1038/s42256-024-00795-w.

[45] C.H. Lampert, H. Nickisch and S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958. doi:10.1109/CVPR.2009.5206594.

[46] F.W. Lawvere, Adjointness in Foundations, *Dialectica* **23**(3/4) (1969), 281–296.

[47] H.J. Levesque, Foundations of a functional approach to knowledge representation, *Artificial Intelligence* **23**(2) (1984), 155–212.

[48] H.J. Levesque, A logic of implicit and explicit belief, in: *Proceedings of the Fourth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Austin, Texas, 1984, pp. 198–202.

[49] M. Lymperaiou and G. Stamou, A survey on knowledge-enhanced multimodal learning, 2023.

[50] S. Mondal, S. Bhatia and R. Mutharaju, EmEL++: Embeddings for EL++ Description Logic, in: *Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, California, USA, March 22-24, 2021*, A. Martin, K. Hinkelmann, H. Fill, A. Gerber, D. Lenat, R. Stolle and F. van Harmelen, eds, CEUR Workshop Proceedings, Vol. 2846, CEUR-WS.org, 2021.

[51] B. Nebel, On the compilability and expressive power of propositional planning formalisms, *Journal of Artificial Intelligence Research* **12** (2000), 271–315.

[52] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen and D. Phung, A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018, pp. 327–333.

[53] Ö.L. Özcep, M. Leemhuis and D. Wolter, Embedding Ontologies in the Description Logic ALC by Axis-Aligned Cones, *Journal of Artificial Intelligence Research* **78** (2023), 217–267.

[54] S.K. Pal and D.P. Mandal, Fuzzy Logic and Approximate Reasoning: An Overview, *IETE Journal of Research* **37**(5–6) (1991), 548–560. doi:10.1080/03772063.1991.11437008.

[55] X. Peng, Z. Tang, M. Kulmanov, K. Niu and R. Hoehndorf, Description Logic EL++ Embeddings with Intersectional Closure, 2022.

[56] S. Powers, *Practical RDF*, O'Reilly Media, 2003, p. 331. ISBN 978-0596002633.

[57] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian Personalized Ranking from Implicit Feedback, in: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, AUAI Press, Arlington, Virginia, USA, 2009, pp. 452–461–. ISBN 9780974903958.

[58] P. Ristoski and H. Paulheim, RDF2Vec: RDF Graph Embeddings for Data Mining, in: *The Semantic Web – ISWC 2016*, P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck and Y. Gil, eds, Springer International Publishing, Cham, 2016, pp. 498–514–. ISBN 978-3-319-46523-4.

[59] M.Á. Rodríguez-García and R. Hoehndorf, Inferring ontology graph structures using OWL reasoning, *BMC Bioinformatics* **19**(1) (2018). doi:10.1186/s12859-017-1999-8.

[60] H.H. Schaefer and M.P. Wolff, *Topological Vector Spaces*, Springer New York, 1999. ISSN 0072-5285. ISBN 9781461214687. doi:10.1007/978-1-4612-1468-7.

[61] K. Schild, A Correspondence Theory for Terminological Logics: Preliminary Report, in: *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, J. Mylopoulos and R. Reiter, eds, Morgan Kaufmann, 1991, pp. 466–471. http://ijcai.org/Proceedings/91-1/Papers/072.pdf.

[62] F.Z. Smaili, X. Gao and R. Hoehndorf, Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations, *Bioinformatics* **34**(13) (2018), i52–i60. doi:10.1093/bioinformatics/bty259.

[63] F.Z. Smaili, X. Gao and R. Hoehndorf, OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction, *Bioinformatics* **35**(12) (2018), 2133–2140. doi:10.1093/bioinformatics/bty933.

[64] L.N. Smith, Cyclical Learning Rates for Training Neural Networks, 2017.

[65] Z. Tang, T. Hinnerichs, X. Peng, X. Zhang and R. Hoehndorf, FALCON: Faithful Neural Semantic Entailment over ALC Ontologies, 2023.

[66] E. van Krieken, E. Acar and F. van Harmelen, Analyzing Differentiable Fuzzy Logic Operators, *Artificial Intelligence* **302** (2022), 103602. doi:https://doi.org/10.1016/j.artint.2021.103602. https://www.sciencedirect.com/science/article/pii/S0004370221001533.

[67] I. Vendrov, R. Kiros, S. Fidler and R. Urtasun, Order-Embeddings of Images and Language, in: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2016.

[68] Y. Venema, 6 Algebras and coalgebras, in: *Handbook of Modal Logic*, P. Blackburn, J. Van Benthem and F. Wolter, eds, Studies in Logic and Practical Reasoning, Vol. 3, Elsevier, 2007, pp. 331–426. ISSN 1570-2464. doi:https://doi.org/10.1016/S1570-2464(07)80009-7. https://www.sciencedirect.com/science/article/pii/S1570246407800097.

[69] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Transactions on Knowledge and Data Engineering* **29**(12) (2017), 2724–2743–. doi:10.1109/TKDE.2017.2754499.

[70] P.L. Whetzel, N.F. Noy, N.H. Shah, P.R. Alexander, C. Nyulas, T. Tudorache and M.A. Musen, BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications, *Nucleic Acids Research* **39**(suppl) (2011), W541–W545–. doi:10.1093/nar/gkr469.

[71] B. Xiong, N. Potyka, T.-K. Tran, M. Nayyeri and S. Staab, Faithful Embeddings for EL++ Knowledge Bases, in: *Proceedings of the 21st International Semantic Web Conference (ISWC2022)*, 2022, pp. 1–18.

[72] B. Yang, W.-t. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, 2015.

[73] F. Zhapa-Camacho and R. Hoehndorf, From axioms over graphs to vectors, and back again: evaluating the properties of graph-based ontology embeddings, 2023.

[74] F. Zhapa-Camacho and R. Hoehndorf, Lattice-Preserving $\mathcal{ALC}$ Ontology Embeddings, in: *Neural-Symbolic Learning and Reasoning*, T.R. Besold, A. d'Avila Garcez, E. Jimenez-Ruiz, R. Confalonieri, P. Madhyastha and B. Wagner, eds, Springer Nature Switzerland, Cham, 2024, pp. 355–369. ISBN 978-3-031-71167-1.

[75] F. Zhapa-Camacho, M. Kulmanov and R. Hoehndorf, mOWL: Python library for machine learning with biomedical ontologies, *Bioinformatics* (2022), btac811. doi:10.1093/bioinformatics/btac811.

Appendix A. The ordered-vector space $(X, \preceq)$

Lemma 1 ($(X, \preceq)$ is a partial order)**.** *The pair* $(X, \preceq)$ *over* $\mathbb{R}^n$*, where for elements* $a, b \in X$ *with* $a = (a_1, ..., a_n)$ *and* $b = (b_1, ..., b_n)$*,* $a \preceq b$ *if and only if* $a_1 \leqslant b_1, ..., a_n \leqslant b_n$*, is a partial order.*

Proof. We demonstrate for each property of a partial order:

1. Reflexivity ($\Rightarrow$): Let $a \in X$ with $a \preceq a$. By definition, we have $a_i \leqslant a_i$ for any i. ($\Leftarrow$): Let $a \in X$. Since $a_i \leqslant a_i$ for any i, then $a \preceq a$.
2. Transitivity ($\Rightarrow$): Let $a, b, c \in X$. If $a \preceq b$ and $b \preceq c$, we have that $a_i \leqslant b_i$ and $b_i \leqslant c_i$; therefore, $a_i \leqslant c_i$ for any i. ($\Leftarrow$): Let $a, b, c \in X$ with $a_i \leqslant b_i$ and $b_i \leqslant c_i$ for any i. It follows that $a_i \leqslant c_i$, which implies $a \preceq c$.
3. Antisymmetry ($\Rightarrow$): Let $a, b \in X$. If $a \preceq b$ and $b \preceq a$, it follows that $a_i \leqslant b_i$ and $b_i \leqslant a_i$. Therefore, $a_i = b_i$ and $a = b$. ($\Leftarrow$): Let $a, b \in X$ with $a_i = b_i$ for any i. It implies that $a_i \leqslant b_i$ and $b_i \leqslant a_i$, therefore, $a \preceq b$ and $b \preceq a$.

□

Appendix B. Hyperparameter Selection

Table 6

Selection of hyperparameters for the different methods with respect to the dataset used. E.S.: Embedding size, L.R.: learning rate, M: margin, B.S.: batch size, N.N.: number of negative samples.

Dataset	Method	E.S.	L.R.	M	B.S.	N.N.
	ELEmbeddings	200	0.0001	0.1	20000	1
GO	Box^{2}EL	200	0.00001	0.1	20000	1
	CatE	200	0.00001	1.0	32768	4
	ELEmbeddings	50	0.001	0.1	20000	1
FoodOn	Box^{2}EL	200	0.0001	0.1	40000	1
	CatE	200	0.0001	1	8192	2
	ELEmbeddings	200	0.00001	0.01	4096	1
ORE1	Box^{2}EL	200	0.0001	0	8192	1
	CatE	200	0.0001	1	32768	4
PPI	CatE	256	0.0001	0.1	2048	4