LPX-AbPoint: Abduction and Pinpointing for Explaining Link Predictions on DL Knowledge Graphs

Roberto Barile[1,*,†], Claudia d'Amato[1,2,*,†] and Nicola Fanizzi[1,2,*,†]

[1]Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Italy
[2]CILA, Università degli Studi di Bari Aldo Moro, Italy

Abstract

The need for explanations to predictions in knowledge graphs is increasing because they are generally computed via embedding-based methods that are efficient, but do not explain predictions. Many explanation methods rely on axioms in the knowledge graph that are incomplete, thus often making the explanations incomplete. One way to address this gap is to consider, for explanations, hypothetical knowledge at the schema or assertion level. However, methods based on hypothetical assertions often ignore existing facts. Conversely, approaches that enrich explanations via rule learning typically disregard the available schema. We propose LPX-AbPoint, the first method using a combination of abduction and pinpointing techniques to compute explanations consisting of both schema axioms and existing or hypothetical assertions. We present a comparative empirical study showing that our method produces more useful explanations than approaches relying solely on observed knowledge, while remaining complementary to rule learning techniques.

Keywords

Knowledge Graphs, Link Prediction, Explanation, Abduction

1. Introduction

Despite their proven utility [1, 2], *Knowledge Graphs* (KGs) [3] are often *incomplete*, due to the their semi-automatic lifecycle. Expressing KGs in *Description Logics* (DLs) [4] allows to exploit their semantics more effectively, based not only on *data assertions* but also on available *schema axioms* via reasoning. Moreover, missing assertions can be also identified through *Link Prediction* (LP). *Knowledge Graph Embedding* (KGE) models [5] are currently widely used for LP tasks as they have proved both accurate and efficient. Essentially they learn mapping KG symbols to vectors (embeddings), thereby encoding structural properties of KGs. Hence, LP tasks can be tackled by means of vector operations. However, these methods can hardly provide evidence (e.g., in the form of axioms) for the predictions. This is an important issue, particularly when predictions may influence critical decisions. For example, predictions on drug side effects can impact investment decisions [6].

To address this issue, we target *LP eXplanation* (LPX) methods [7], focusing specifically on *post-hoc* ones as they can be combined with any LP method. Many such methods, e.g., Kelpie [8], leverage the embeddings to compute explanations based on assertions available in the KG. However, these methods discard the available schema. Alternatively, *pinpointing* methods [9] may be used to explain predictions that are logically entailed by the KG. The resulting explanations also include schema axioms, which are crucial for explanation [10]. However, KGs do not always entail the predictions computed via KGE models, also because of their inherent incompleteness. This often leads LPX methods based on the embeddings to return explanations that do not clearly entail the prediction and pinpointing methods to fail to explain the prediction. Our aim is to overcome these limitations and improve the *utility* of explanations. To this purpose, we explore the usage of additional knowledge that is missing from the KG but possibly true, according to the *open world semantics* adopted in DLs, and useful for explanation.

DL 2026: 39th International Workshop on Description Logics, July 17–19, 2026, Lisbon, Portugal

*Corresponding author.
†These authors contributed equally.
r.barile17@phd.uniba.it (R. Barile); claudia.damato@uniba.it (C. d'Amato); nicola.fanizzi@uniba.it (N. Fanizzi)
https://rbarile17.github.io/robertobarile.github.io/ (R. Barile)
0009-0007-3058-8692 (R. Barile); 0000-0002-3385-987X (C. d'Amato); 0000-0001-5319-7933 (N. Fanizzi)

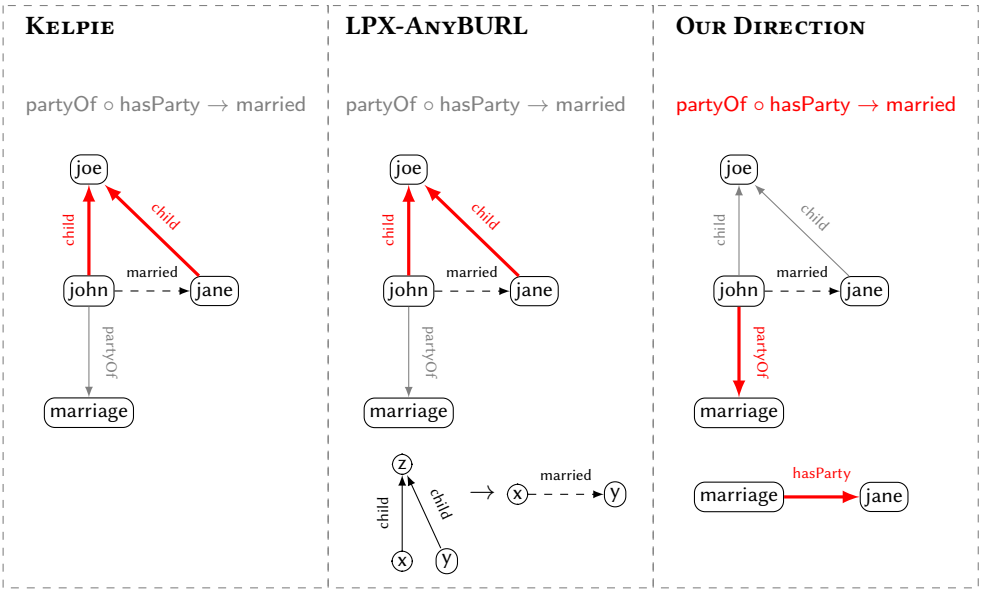

Figure 1: Comparison of three explanations for a prediction. The dashed edges denote predictions. The thick red (thin gray) elements denote the knowledge used (not used) in the explanation.

Existing LPX methods that use additional assertions rely on such supplementary knowledge even when the available knowledge is sufficient. Also, they disregard schema axioms. As an alternative, LPX-AnyBURL explains predictions inductively: it selects learned rules, along with their grounding assertions, to lead to the prediction. However, in this approach the rules are expressed in a different language from that of the KG schema, which is therefore discarded.

As a complementary direction, we investigate the use of additional assertions in combination with available knowledge. Specifically, we explore explanations consisting of: 1) schema axioms; 2) existing assertions, additional assertions, or both. As an example, consider the explanations, compared in Fig. 1, for the same prediction on a KG, with data and schema, about kinship relations. Kelpie may return the child-assertions as an explanation. However, this explanation does not clearly entail the prediction even if coupled with the available schema. LPX-AnyBURL, instead, may explain the prediction using the same childassertions and a learned rule stating that two persons sharing a child are (probably) married. Differently, an additional assertion could regard the other participant of the marriage, which would lead to also use the (available) schema axiom.

To generate the additional assertions, we resort to *abduction* to generate a *hypothesis*, i.e., an assertion that helps entail the prediction. MHS-MXP [11] is a abduction method that does not impose any language restrictions and is based on the interpretations of the KG that do not satisfy the prediction. However, the inspection of interpretations is not implemented in recent parallel DL reasoners (e.g., Konclude [12]) that scale to large KGs. We address this limitation to allow explanations to be computed through abduction exploiting efficient reasoners.

For our purposes, we propose LPX-AbPoint, which, to the best of our knowledge, is the first post-hoc LPX method to compute explanations with such a structure, combining *abduction* and *pinpointing* [13]. LPX-AbPoint resorts to abduction to generate a hypothesis and performs pinpointing on the KG extended by assuming the hypothesis. The abduction method applies to any DL with decidable consistency checking, whereas the pinpointing approach is tailored to tableau reasoning, though other approaches can be integrated. The abduction approach further generalizes MHS-MXP by characterizing hypotheses via the known duality between unsatisfiable and correction constraint sets [14], which only requires decidable consistency verification via any reasoner. As for pinpointing, LPX-AbPoint relies on tracing extensions of reasoning because that have the same complexity as standard reasoning.

We posit that LPX-AbPoint, by using abduction as an intermediate step to frame LPX as pinpointing, yields higher utility than methods using only available knowledge and those using only additional

assertions and performs comparably to LPX-AnyBURL. We tested this claim through a comparative evaluation on multiple datasets derived from large and publicly available KGs [15] and using LP-diXit [16], a proxy indicator of explanation utility based on LLMs [17].

The contributions of this work are the following:

- we apply abduction and pinpointing to explain predictions by LP methods;
- we apply the known duality between unsatisfiable sets and correction constraint sets to abduction in DLs;
- we experimentally prove the utility of abduction for explaining predictions by LP methods.

The rest of the paper is organized as follows. We review current solutions for LPX in §2, while basics on KGE are provided in §3. We describe LPX-AbPoint in detail in §4 and present an empirical study on its implementation in §5. Finally, we summarize the contributions and suggest possible extensions in §6.

2. Related Works

In this section, we survey state-of-the-art approaches to explain LP in KGs. Among *post-hoc* LPX methods, that we target because they are general w.r.t. the LP method, several ones select or generate pieces of knowledge as evidence for the given prediction. The simplest methods in this class compute exactly one assertion by optimizing for a measure of the explanatory utility of possible assertions. This is the case of DP [18], using a perturbation measure, and Criage [19] based on approximated influence functions. Criage targets assertions sharing the second argument with the prediction and applies to geometrical and tensor factorization KGE models, whereas DP targets assertions sharing the first argument with the prediction and applies to any KGE model. More advanced methods explain a prediction by returning a set of assertions selected w.r.t. to an objective measure among a candidates pool. Kelpie [8] and Kelpie++ [20] frame the objective as a partial re-training process. Similarly, KE-X [21] is based on (approximated) information gain, whereas KGExplainer [22] and KGEx [23] are based on knowledge distillation. CrossE [24] and SemanticCrossE [25] generate explanations consisting of paths. They are based on similarity measures and support paths, i.e., similar predictions with a similar explanation. Although they apply in principle to any KGE model, their performance largely depends on the usage of specific models that also encode interactions among individuals and properties. However, these methods rely only on knowledge observed in the KG, whereas we target explanations enriched with additional knowledge. Alternatively, *pinpointing* methods [9] may be used to explain predictions that are logically entailed by the KG. The resulting explanations may also include schema axioms, which are a crucial component of explanation [10]. However, this approaches fail to explain predictions that are not entailed by the KG.

Imagine, which is also based on partial re-training, targets hypothetical assertions computed via graph summarization and subsequent reconstruction of the original graph. Kelpie and Kelpie++ involve additional assertions obtained by corrupting possible explanations, but return solely the existing assertions, from which the additional ones are obtained. Imagine, instead, provides explanations consisting of actual additional assertions. However, it discards the available assertions even when useful for explanation. Instead, we aim for an approach considering hypothetical or existing assertions. Moreover, while intuitively leading to "plausible" hypotheses, Imagine is not grounded in any uncertainty or reasoning criterion. We rather aim for a principled account of hypotheses based on their ability to support entailment. GEnI [26] also includes schema axioms in the explanations. It infers schema axioms by matching numerical patterns in property embeddings to schema axioms on properties (e.g., two properties are inverse when their embeddings sum to zero). However, these patterns are seldom observed in practice: GEnI falls back to an approach similar to Criage, when no match is found. Moreover, GEnI works only for geometrical and tensor factorization KGE models.

These methods often apply to any LP method, but are dependent on the embeddings. Instead, LPX-AnyBURL [27] is fully independent of the embeddings: it learns rules from the observed assertions and selects a subset of rules and facts that lead to the prediction. It frames this step as an abduction

problem: starting from the rules (i.e., assuming no assertions), it identifies which assertions must be added to derive the prediction. In this approach, the candidate hypotheses coincide with the existing facts in the KG. This is essentially equivalent to rule pinpointing over the KG. In contrast, we aim at an abduction approach where the possible hypotheses are assertions not observed in the KG.

Other *post-hoc* methods target explanation more globally: they aim to construct clear models from the original opaque models and/or their predictions rather than return knowledge that back up a specific prediction. For example, in [28] logical rules are mined to explain a set of predictions. With FeaBI [29], interpretable vectors are extracted from embeddings (of KGE models) via feature engineering. Another direction is represented by interpretable LP methods, e.g., interpretable embeddings [30] or rule-based LP methods [31, 32], where the prediction processes/mechanisms are more traceable.

As for abduction in DLs, several methods have been formalized targeting the extension of data [33, 34], schema [35], both data and schema [36], or complex concept expressions [37]. Among the assertion abduction methods that we target, MHS-MXP [11], based on the seminal HS-tree method [38], is the most general in that it does not impose any language restriction. However, it requires the inspection of DL interpretations which is seldom implemented in DL reasoners. Also recursive partitioning algorithms such as QuickXplain and MXP [39] can be used to derive minimal hypotheses. In particular, while QuickXplain [40, 41] returns one minimal hypothesis, MXP computes all hypotheses of minimum cardinality and, when larger ones exist, returns one of them. Different hypotheses can be obtained by changing the (often arbitrary) partitioning approach. However, these methods do not guarantee that the returned hypotheses are consistent with the KG.

3. Basics

First, we specify the DL notation adopted in the paper.[1] Then, we recall the basics of KGE models.

Definition 1 (DL KG). *Let* $\mathbf{I}$, $\mathbf{C}$, *and* $\mathbf{R}$ *be the sets of individual names, concept names (including* $\top$ *and* $\bot$*), and property names, respectively. For concepts* $A, B \in \mathbf{C}$*, the concept intersection is the concept* $A \sqcap B$*. A DL KG is a pair* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, where* $\mathcal{T}$ *is a TBox and* $\mathcal{A}$ *is an ABox. For* $a, b \in \mathbf{I}$*,* $C \in \mathbf{C}$*, and* $r \in \mathbf{R}$*, concept assertions are expressions of the form* $C(a)$ *and property assertions are expressions of the form* $r(a, b)$*. For concepts* $A, B \in \mathbf{C}$*, concept subsumption axioms are expressions of the form* $A \sqsubseteq B$*.*

KGE models typically learn a representation of each individual, concept, and property name in the KG as an embedding (vector) in a low-dimensional space by exploiting the ABox [5]:

Definition 2 (KGE model). *Let* $d_i, d_r \in \mathbb{N}$ *be the embedding dimensions for individual names and property names, respectively. A KGE model represents each individual name* $a \in \mathbf{I}$ *and each concept name* $C \in \mathbf{C}$ *such that a concept assertion* $C(a)$ *is in* $\mathcal{A}$ *by means of an embedding* $\mathbf{a} \in \mathbb{R}^{d_i}$*; and each property name* $r \in \mathbf{R}$ *and a* $type$ *property dedicated to concept assertions through an embedding* $\mathbf{r} \in \mathbb{R}^{d_r}$*.*

Moreover, KGE models involve a score function that, given the embeddings, estimates the plausibility of assertions. Different categories of KGE models, such as geometrical, based on tensor factorization, and neural ones, are often characterized by the adopted score function. The LP methods compute the filler for an incomplete triple by ranking the possible triples based on the score function.

4. The Proposed Approach

We introduce the *post-hoc* LPX method LPX-AbPoint. It can be combined with any LP method and is generally independent of the embeddings, i.e., the explanations it produces do not vary with the underlying KGE model. It takes a consistent KG $\mathcal{K}$ and a ABox assertion p predicted via a LP method; we define $\mathcal{K}' \triangleq \mathcal{K} \cup \{\neg p\}$. It computes an explanation consisting of:

[1] We define notation only for the concept constructors and TBox axioms that appear in the examples because the formalization of LPX-AbPoint is independent of the specific notation of these constructs.

- a set of existing TBox axioms;
- a (possibly empty) set of existing ABox assertions;
- a (possibly empty) hypothesis.

LPX-AbPoint uses ABox abduction to identify the hypothesis. It then employs axiom pinpointing to extract the explanation (justification) from the KG extended by assuming the hypothesis. The abduction method requires only a decidable consistency checking procedure (thus also applying to formalisms beyond DLs), whereas the axiom pinpointing approach that we adopt is tailored to tableau reasoning, even if in principle other approaches can be integrated. In the remainder of this section, we detail our abduction approach (§4.1) and the integration of axiom pinpointing approaches (§4.2).

4.1. ABox Abduction in LPX-AbPoint

In this subsection, we characterize a hypothesis via KG consistency, illustrate the method proposed to compute a hypothesis based on this characterization, and discuss the computational complexity of this method.

4.1.1. Hypotheses and Correction Sets

A hypothesis is a set of ABox assertions, disjoint from the KG ABox, that, if assumed, leads to entail the prediction. A hypothesis is trivial if it is inconsistent with the KG, as a contradiction entails any assertion. Formally:

Definition 3 (Hypothesis)**.** *Let $\mathcal{U}$ be a finite set of ABox assertions called abducibles that is disjoint from $\mathcal{A}$, a hypothesis $\mathcal{H} \subseteq \mathcal{U}$ for p given $\mathcal{K}$ is a set of abducibles that if assumed leads to entail the prediction, i.e., $\mathcal{K} \cup \mathcal{H} \models p$ ($\mathcal{K}' \cup \mathcal{H}$ is inconsistent). A hypothesis $\mathcal{H}$ is minimal iff there is no other hypothesis $\mathcal{H}'$ such that $\mathcal{H}' \subset \mathcal{H}$. A hypothesis $\mathcal{H}$ is* non-trivial *(*trivial*) if $\mathcal{K} \cup \mathcal{H}$ is consistent (inconsistent).*

Example 1. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KG with $\mathcal{A} = \{E(a)\}$ (shortly $\{E\}$), $\mathcal{T} = \{\, E \sqcap A \sqcap C \sqsubseteq S,\ \ E \sqcap B \sqcap D \sqsubseteq S,\ \ F \sqcap G \sqsubseteq \bot \,\}$, and enriched with the prediction $S(a)$, and $\mathcal{U} \triangleq \{A(a),\, B(a),\, C(a),\, D(a)\}$ (shortly $\{A, B, C, D\}$) be the set of abducibles. The minimal non-trivial hypothesis are $\{A, C\}$ and $\{B, D\}$. Instead, $\{F, G\}$ is a minimal trivial hypothesis.*

Since the definition does not yield an efficient procedure to compute non-trivial minimal hypotheses, other than testing for all subsets of abducibles, we further characterize a hypothesis via a duality known for sets of constraints [14]. In particular, minimal hypotheses correspond to *Minimal Unsatisfiable Subsets* (MUSes) of $\mathcal{U}$, i.e., minimal subsets of $\mathcal{U}$ whose addition to $\mathcal{K}'$ makes it inconsistent. MUSes are dual to *Minimal Correction Subsets* (MCSes) of $\mathcal{U}$, i.e., minimal subsets of $\mathcal{U}$ whose removal from $\mathcal{K}' \cup \mathcal{U}$ makes it consistent. The duality is characterized via *Hitting Sets* (HSes). Given the family $\mathcal{F}$ of MCSes, a HS is a set H that hits all the sets in $\mathcal{F}$, i.e., $H \cap F \neq \emptyset$ for each $F \in \mathcal{F}$. A *Minimal HS* (MHS) is a HS minimal w.r.t. set inclusion. This duality is formalized in the following theorem:

Theorem 1. *$\mathcal{H}$ is a minimal hypothesis for $\mathcal{K}'$ (a MUS of $\mathcal{K}'$) iff $\mathcal{H}$ is a MHS of the MCS family $\mathcal{F}$.*

Proof. We prove that $\mathcal{H}$ is a hypothesis (i.e., $\mathcal{K}' \cup \mathcal{H}$ is inconsistent) iff $\mathcal{H}$ is a HS of $\mathcal{F}$ by proving the two implications separately, and then prove minimality.
($\Rightarrow$). Let $F \in \mathcal{F}$, suppose by contradiction that $\mathcal{H}$ does not hit F, i.e., $\mathcal{H} \cap F = \emptyset$. Since $\mathcal{H} \subseteq \mathcal{U}$, this implies $\mathcal{H} \subseteq (\mathcal{U} \setminus F)$ and therefore $\mathcal{K}' \cup \mathcal{H} \subseteq \mathcal{K}' \cup (\mathcal{U} \setminus F)$. By definition of $\mathcal{F}$, $\mathcal{K}' \cup (\mathcal{U} \setminus F)$ is consistent. However, this implies, by monotonicity, that $\mathcal{K}' \cup \mathcal{H}$ is consistent, contradicting the assumption. Therefore, $\mathcal{H}$ is a HS of $\mathcal{F}$.
($\Leftarrow$). Suppose by contradiction that $\mathcal{K}' \cup \mathcal{H}$ is consistent. Then, by definition of $\mathcal{F}$, we have $\mathcal{U} \setminus \mathcal{H} \in \mathcal{F}$. However, $\mathcal{H}$ does not hit this member of $\mathcal{F}$ ($\mathcal{H} \cap (\mathcal{U} \setminus \mathcal{H}) = \emptyset$), contradicting the assumption. Therefore, $\mathcal{K}' \cup \mathcal{H}$ is inconsistent.
Minimality. If $\mathcal{H}$ is a MHS of $\mathcal{F}$, then no strict subset of $\mathcal{H}$ is a HS of $\mathcal{F}$; hence, no strict subset of $\mathcal{H}$ is a hypothesis, and thus $\mathcal{H}$ is a minimal hypothesis. Conversely, if $\mathcal{H}$ is a minimal hypothesis, then no strict subset of $\mathcal{H}$ is a hypothesis; hence, no strict subset of $\mathcal{H}$ is a HS of $\mathcal{F}$, and thus $\mathcal{H}$ is a MHS. □

Example 2. *The MCSes family (of $\mathcal{K} \cup \{\neg S(a)\} \cup \mathcal{U}$) is $\mathcal{F} = \{\{A, B\}, \{A, D\}, \{C, B\}, \{C, D\}\}$. The MHSes of $\mathcal{F}$ are $\{A, C\}$ and $\{B, D\}$. Each MHS leads to entail the prediction: $\mathcal{K} \cup \{A, C\} \models S(a)$ and $\mathcal{K} \cup \{B, D\} \models S(a)$.*

4.1.2. Computing a Minimal Non-Trivial Hypothesis

LPX-AbPoint computes the first non-trivial minimal hypotheses by enumerating the minimal hypotheses until a non-trivial one is found. Particularly, LPX-AbPoint selects a set of abducibles and enumerates the MHSes of the MCSes of the selected abducibles.

Selecting the Abducibles LPX-AbPoint includes a prior step for selecting abducibles because, for very large KGs, considering as abducibles all the assertions missing in the KG (typically more than the existing assertions) is clearly computationally infeasible. Hence, it selects the abducibles that are most likely to belong to a hypothesis. Specifically, it performs a heuristic selection based on embedding similarity between the prediction and the possible abducibles.

Definition 4 (Abducibles)**.** *Given a prediction $r(s, o)$ and a parameter k, the set of abducibles $\mathcal{U}$ contains:*

i. the top-k concept assertions of the form $A(s)$ or $A(o)$ that are not in $\mathcal{K}$ and differ from the prediction, ranked according to the cosine similarity between the embeddings $\mathbf{A}$ and $\mathbf{s}$ or $\mathbf{o}$;

ii. the top-k role assertions of the form $q(s, o)$ or $q(o, s)$ that are not in $\mathcal{K}$ and differ from the prediction, ranked according to the cosine similarity between the embeddings $\mathbf{q}$ and $\mathbf{r}$, with arbitrary tie-breaking between $q(s, o)$ and $q(o, s)$.

This heuristic selection represents a practical trade-off: although it does not guarantee that all relevant abducibles are retained, it is a efficient criterion to prioritize assertions related to the prediction.

Enumerating the Minimal Hitting Sets LPX-AbPoint grounds on (a slight variation of) the HS-tree method in MHS-MXP for computing the MHSes of $\mathcal{F}$.

Definition 5 (HS-tree)**.** *An HS-tree of $\mathcal{F}$ is a node-labeled and edge-labeled tree. For each node n, let $H(n)$ denote the set of edge labels occurring on the path from the root to n. The node label $L(n)$ can be either: an MCS $S \in \mathcal{F}$ such that $S \cap H(n) = \emptyset$; or ✓, if no such MCS exists, which means that $H(n)$ is a HS of $\mathcal{F}$. If $L(n)$ is an MCS S, then for every element $\sigma \in S$ there is a child n_σ of n s.t. the edge from n to n_σ is labeled with σ.*

LPX-AbPoint adopts an algorithm for building a HS-tree of the family $\mathcal{F}$ of MCSes where the paths to the nodes labeled with ✓ are the MHSes of $\mathcal{F}$. Differently from MHS-MXP, in this algorithm the nodes are labeled by computing MCSes rather than by inspecting DL interpretations. Since computing MCSes is computationally demanding, the algorithm treats the family $\mathcal{F}$ as an implicit set, computing the elements as needed and trying to minimize the number of required labels. Indeed, it builds the HS-tree in breadth-first order, uses two pruning rules, and adopts a MCS reusing strategy.

The following describes the algorithm in detail. It adopts a queue of nodes initialized with the root node, which is labeled with a MCS of $\mathcal{K}' \cup \mathcal{U}$. For each node n in the queue, it first checks the following pruning conditions: (1) if another node n' is labeled with ✓ and $H(n') \subseteq H(n)$, node n is pruned in order to stop exploring a superset of a MHS that is obviously non-minimal; (2) if $H(n) = H(n')$ (equal up to permutation as paths are sets) for another node n', then n is pruned because it would lead to an identical branch. Then, if the node is not pruned, the algorithm checks if it leads to a hypothesis, i.e., $\mathcal{K}' \cup H(n)$ is inconsistent. If so, the prediction is entailed and $H(n)$ is returned if it is a non-trivial hypothesis, i.e., $K \cup H(n)$ is consistent, or the branch is pruned otherwise. If $H(n)$ does not lead to entail the prediction, a label is needed for n. The algorithm first adopts a reuse strategy: if there exists a node n' such that $L(n') \cap H(n) = \emptyset$ then $L(n')$ is reused as the label of n. Otherwise, it computes $L(n)$ as a MCS of $\mathcal{K}' \cup \mathcal{U}$ using $\mathcal{U} \setminus H(n)$ as possible MCS elements. A different MCS is obtained as the set of possible MCS elements is restricted. It then adds each element of $L(n)$ to the queue.

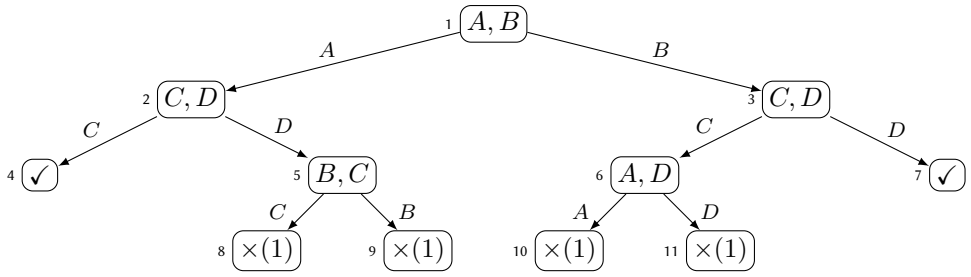

Figure 2: HS-tree example. Nodes are numbered for reference in text.

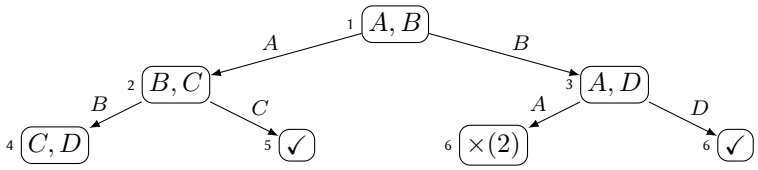

Figure 3: HS-tree example. Nodes are numbered for reference in text.

Example 3. *We describe the computation of the HS-tree illustrated in Fig. 2. At the root, the MCS is* $\{A, B\}$*: the algorithm branches on* A *and* B*. The KG is still consistent if adding* $H(2) = \{A\}$*, hence, node 2 is labeled with the MCS* $\{C, D\}$*. For node* 3*, the algorithm reuses the label of node* 2 *because its path* $H(3) = \{B\}$ *is disjoint with the label of node* 2*. Node* 4 *is labeled* ✓ *as adding the path* $H(4) = \{A, C\}$ *to the KG leads to entail the prediction. While for node* 5 *and node* 6 *a label is computed, node* 7 *is labeled* ✓ *as adding* $H(7) = \{B, D\}$ *leads to entail the prediction. At the next level, node* 8 *is pruned by condition (1) because* $H(8) = \{A, D, C\}$ *is a superset of the MHS* $\{A, C\}$*. Similarly, nodes* 9, 10*, and* 11 *are pruned by condition (1).*

Example 4. *We describe the computation of the HS-tree illustrated in Fig. 3. The root is labeled with* $\{A, B\}$*. Node* 2 *is labeled with the MCS* $\{B, C\}$*. Similarly, node* 3 *is labeled with the MCS* $\{A, D\}$*. At the next level, node* 6 *is pruned by condition (2) because its path* $H(6) = \{B, A\}$ *is equal to the path to node 4, namely* $H(4) = \{A, B\}$*.*

Finding a Minimal Correction Set For finding a MCS of the selected abducibles, LPX-AbPoint resorts to a novel instantiation of QuickXplain [40, 41] (QX). It is a "*divide et impera*" algorithm that recursively partitions the input set to find a minimal subset satisfying a given monotone property. LPX-AbPoint applies QX to compute a MCS F by instantiating the monotone property as the consistency of $\mathcal{K}' \cup (\mathcal{U} \setminus F)$, which is monotone (although consistency is anti-monotone) as adding axioms to F means removing axioms from $\mathcal{K}' \cup \mathcal{U}$. QX solely requires a decidable consistency verification procedure. Note that, for a fixed partitioning of the input set, QX always returns the same subset, while different partitions may yield different subsets.

We formalize this instantiation of QX in Alg. 1 and describe its initial call and first recursion level. QX takes as input a finite set U of possible solutions; a background set B against which U is evaluated; a removal set R, initially empty, tracking the elements removed from B during recursion. To find a MCS $F \subseteq \mathcal{U}$ of $\mathcal{K}' \cup \mathcal{U}$, LPX-AbPoint invokes QX with $U = \mathcal{U}$, $B = \mathcal{K}' \cup \mathcal{U}$ (after checking that $\mathcal{K}' \cup \mathcal{U}$ is inconsistent because otherwise no hypothesis exists), and $R = \emptyset$.

The first condition (Line 1) fails as the removal set R is initially empty. If U contains a MCS (guaranteed by the initial check) and it has only one element (Line 4), then U is a MCS (Line 5). Otherwise, QX partitions U into two disjoint subsets U_1 and U_2 (Line 7). Initially, it finds in the first subset U_1 a MCS F_1 of the background without the second subset $B \setminus U_2$ (Line 8). Then, it finds in the second subset U_2 a MCS F_2 of the background without the first part of the MCS $B \setminus F_1$ (Line 9). Eventually, it returns $F = F_1 \cup F_2$ (Line 10). In the left recursion, the removal set is non-empty as this call is performed when U contains more than one element. Hence, QX determines whether to discard U_1 or continue

Algorithm 1 QX

Input: Solutions set U, background B, removal set R.
Output: A minimal subset $F \subseteq U$ that is a minimal MCS of B.

1: **if** $R \neq \emptyset \ \wedge \ \text{Consistent}(B)$ **then**
2: **return** $\emptyset$
3: **end if**
4: **if** $|U| = 1$ **then**
5: **return** U
6: **end if**
7: Choose U_1, U_2 s.t. $U = U_1 \cup U_2$ and $U_1 \cap U_2 = \emptyset$
8: $F_1 \leftarrow \text{QX}(U_1,\ B \setminus U_2,\ U_2)$
9: $F_2 \leftarrow \text{QX}(U_2,\ B \setminus F_1,\ F_1)$
10: **return** $F_1 \cup F_2$

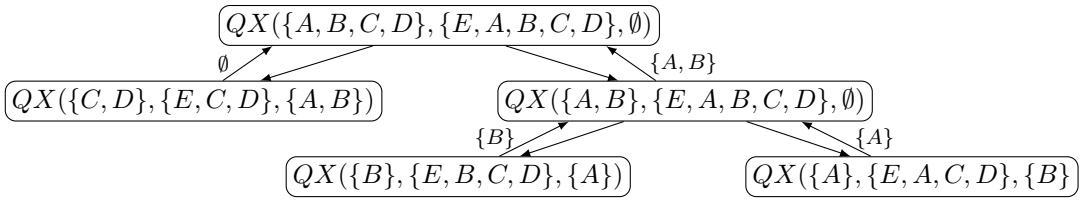

Figure 4: Recursion tree for a execution of QX: downward edges denote recursive calls, nodes denote call arguments (in the second argument, the TBox $\mathcal{T}$ is omitted because it is not modified across calls), upward edges denote returned values.

analyzing it: if the background without U_2 is already consistent (Line 1), a MCS must lie in U_2 and QX discards U_1 (Line 2); otherwise, U_1 contains an element of a MCS. Next, if U_1 has only one element (Line 4), QX returns it (Line 5); otherwise QX partitions U_1 (Line 7) and performs the recursive calls (Line 8 and Line 9). In the right recursion, if F_1 is empty, the consistency check can be skipped, as U_2 must contain elements of a MCS (Line 1). Otherwise, QX determines whether to discard U_2 or continue analyzing it: if the background without the already identified part of the MCS F_1 is consistent (Line 1) it discards U_2 (Line 2); otherwise, some other elements of a MCS are in U_2. The subsequent steps either lead to return U_2 (Line 5) or continue with further recursive calls (Line 8 and Line 9).

Example 5. *We describe the recursion tree illustrated in Fig. 4 for an execution of QX computing a MCS for Ex. 1. Since the removal set is initially empty and $\mathcal{B}$ is inconsistent, the set of possible solutions is partitioned into $\{C, D\}$ and $\{A, B\}$. In the left recursion, QX discards $\{C, D\}$ as removing $\{A, B\}$ restores consistency. In the right recursion, it further splits $\{A, B\}$ into $\{B\}$ and $\{A\}$ as the MCS portion identified in the left recursion is empty and the solutions set is not a singleton. For the first split, the singleton $\{B\}$ is returned because removing A is insufficient. The same happens for the second split, where the singleton $\{A\}$ is returned. QX eventually returns $\{A, B\}$.*

4.1.3. Computational Complexity of Abduction in LPX-AbPoint

In the following, we discuss the complexity of computing the first minimal hypothesis, and for enumerating minimal hypotheses, which is the problem addressed by LPX-AbPoint. The complexity results are stated w.r.t. the number of consistency checks, treating each one as a constant time operation.

Proposition 1. *The following results hold [42]:*

i. Computing the first minimal hypothesis is in P.
ii. Computing the next minimal hypothesis is NP-complete. Restricting to hypotheses of size at most k, the problem is in P. Precisely, in the worst case, $\mathcal{O}(n^{k+1})$ consistency checks are required, where n is the number of abducibles.

Proof sketch. We prove the two results separately:

i. By monotonicity of entailment, the first minimal hypothesis can be obtained by starting from all the abducibles and iteratively removing them while entailment of the prediction is preserved.
ii. LPX-AbPoint constructs the HS-tree in breadth-first order up to depth k. Hence, it requires up to $\mathcal{O}(n^k)$ node expansions. Each node is labeled via QX, which requires up to $\mathcal{O}(n)$ consistency checks. Overall, up to $\mathcal{O}(n^{k+1})$ consistency checks are performed. □

4.2. Integrating Axiom Pinpointing Methods in LPX-AbPoint

We define an explanation and motivate the approach integrated in LPX-AbPoint.

Definition 6 (Explanation)**.** *Given a KG $\mathcal{K}$ and an assertion p such that $\mathcal{K} \models p$, an explanation for p is a subset $\mathcal{J} \subseteq \mathcal{K}$ such that $\mathcal{J} \models p$. An explanation is minimal if no proper subset $\mathcal{J}' \subset \mathcal{J}$ entails p.*

Example 6. *The minimal explanations for the KG in Ex. 1, extended by assuming the hypothesis, are $\{E, A, C\}$ and $\{E, B, D\}$.*

To compute an explanation of p given $\mathcal{K} \cup \mathcal{H}$, LPX-AbPoint resorts to axiom pinpointing approaches that compute explanations based on *tracing* of the reasoning via the tableau algorithm, which are also known as *glass-box* approaches [13]. These methods return explanations consisting of both TBox and ABox axioms and apply to very expressive DLs such as $\mathcal{SROIQ}$. They tag the assertion of the ABox(es) created during the expansion steps in the tableau with the axioms in the KG that led to such expansions. An obvious drawback of these pinpointing methods is that the tracing mechanism requires to disable the main optimizations of the tableau method, i.e., to stop exploring alternative expansions once a clash is found. For mitigating this issue, we focus on returning only one explanation. Hence, this method falls within the same complexity as the tableau without tracing, which depends on the targeted DL. Moreover, termination is guaranteed when returning only one explanation, whereas tracing extensions for retrieving all the explanations may not terminate (even if the tableau without tracing terminates). However, the set of axioms generated by this algorithm might not be minimal. The redundant axioms may be caused by the order of tableau rule applications.

Differently, *black-box* approaches for explanation do not require modifications to the underlying reasoning algorithm, but, even for computing only one explanation, require multiple consistency checks. The *glass-box* pinpointing approach can be combined with a *black-box* one for reducing the non minimal explanation to a minimal one.

5. Experimental Evaluation

In this section, we discuss the settings (§5.1) and the outcomes (§5.2) of the comparative empirical evaluation that we performed.

5.1. Experimental Setting

We used LP-diXit [16] to measure the utility of the returned explanations. LP-diXit measures the *Forward Simulatability Variation* (FSV) induced by an explanation for a prediction (made by a LP method), i.e., the variation between the simulatability (predictability) of a prediction without and with an explanation [43]. A prediction (with an explanation) is simulatable if a verifier can hypothesize the output of the LP method given the same input provided to the (LP) method (and the explanation). The FSV can have the following values, indicating that the explanation is:

$+1$ *useful*: the simulation with the explanation is correct, whereas the one without it is incorrect;
0 *useless*: either both simulations are correct or both are incorrect;
-1 *misleading*: the simulation with the explanation is incorrect, instead, the one without it is correct.

We adopted LP-diXit with the LLM Llama3.1-70B-Instruct using a *zero-shot prompt* including a constraint forcing the LLM to select its response from the top 20 fillers of the prediction input, according to the KGE model. As metrics, we measured the frequencies f_{+1}, $f_{\pm 0}$, and f_{-1} of the corresponding FSV values, and the average time (t) required to compute an explanation.

We configured LPX-AbPoint to stop the construction of the HS-tree as soon as a minimal non-trivial hypothesis is found. Moreover, we set both the maximum depth of the HS-tree and the parameter k in the selection of abducibles to 2 because this is the largest computationally feasible configuration. Under this setting, LPX-AbPoint requires up to 12^3 consistency checks, which corresponds to ~ 9 hours of computation time, if considering ~ 20 seconds per consistency check.

We compared LPX-AbPoint with several recent LPX methods producing different types of explanations: Criage and DP (single assertions), Kelpie++ (sets of assertions), Imagine (hypothetical assertions), and LPX-AnyBURL (rule-based explanations). For Kelpie++ and Imagine, we set the maximum explanation length to 2 because longer explanations were shown to be less useful despite the higher computational cost [16]. Although Kelpie++ supports both necessary and sufficient explanation modalities and two different graph summarization approaches (simulation and bisimulation), we adopt only the necessary modality and the summarization based on simulation, since the other alternatives were found to be less effective, while decreasing efficiency [16]. We adopted four datasets collected in KG-SaF [15], a recent suite of datasets extracted from publicly available KGs, containing both TBox and ABox, and released in `OWL2 DL`. Specifically, we selected *YAGO3-10-C*, *YAGO4-20-C*, *ARCO25-5*, *WHOW25-5* (excluding DBPEDIA25-100K-C as it is inconsistent). These cover different levels of expressivity in the schema, with ARCO25-5 and WHOW25-5 being the most expressive. For each model and dataset, we compared the LPX methods on a sample of 100 assertions among those correctly top ranked by the LP method. Since the explanations (by some LPX methods) may vary depending on the LP method, we performed the experiments with different LP methods, each representing a prominent model family, namely: TransE [44] (*geometrical*), ConvE [45] (*neural*) and ComplEx [46] (*tensor factorization*). We executed Criage solely for ComplEx and ConvE because it does not apply to translational models.

We used Konclude, a hybrid (tableau and consequence-based) parallel `OWL2 DL` reasoner, for consistency checking, and Pellet [47] for axiom pinpointing, as it is, to the best of our knowledge, the only reasoner supporting glass-box axiom pinpointing. We implemented this study in `Python` on the software libraries `GRainsaCK` [48], that automatizes the evaluation of LPX methods via LP-diXit, and `PyClause` [49], implementing LPX-AnyBURL. The resources require to reproduce the study, e.g., code, prompts, required libraries, adopted hardware, datasets, trained models, and hyperparameters, are available on `GitHub`.[2]

5.2. Outcomes and Discussion

Tab. 1 shows that methods leveraging additional knowledge (LPX-AbPoint and LPX-AnyBURL) consistently outperform those relying solely on observed knowledge (Criage, DP, Kelpie++) or on additional assertions (Imagine), supporting the claim that explaining predictions requires additional knowledge when existing knowledge is insufficient, while still reusing existing knowledge whenever it suffices. One possible reason leading LPX-AnyBURL to outperform LPX-AbPoint is that it relies on a large set of inductively learned rules that, even if uncertain, account for most predictions. By contrast, schema axioms in ontologies are not designed to predict all the observed assertions. Hence, LPX-AbPoint requires either that the prediction is already entailed by the ontology or that the missing gap can be bridged through abduction. A further observation concerns the variability of the performance of LPX-AbPoint across datasets. LPX-AbPoint obtained the highest fraction of helpful explanations on YAGO3-10-C, followed by ARCO25-5, YAGO4-20-C, and WHOW25-5. It may be partly due to ARCO25-5 and WHOW25-5 being highly domain-specific KGs where many individuals have technical identifiers that may have reduced the LLM's ability to use the explanations. As for efficiency, although the results highlight limitations of LPX-AbPoint, it is, to the best of our knowledge, the first method applying

[2] https://github.com/rbarile17/lpx-abpoint

Table 1
The FSV value frequencies and the average explanation time (s) of the LPX methods.

KGE	LPX method	YAGO3-10-C				YAGO4-20-C				ARCO25-5				WHOW25-5			
		f_{+1}	$f_{\pm 0}$	f_{-1}	t	f_{+1}	$f_{\pm 0}$	f_{-1}	t	f_{+1}	$f_{\pm 0}$	f_{-1}	t	f_{+1}	$f_{\pm 0}$	f_{-1}	t
ComplEx	Criage	70	30	0	0	29	67	4	0	44	55	1	0	47	52	1	0
	DP	28	72	0	0	10	80	10	0	11	84	5	0	12	84	4	0
	Imagine	3	94	3	151	13	76	1	248	13	81	6	128	15	81	4	56
	Kelpie++	44	55	1	168	12	80	8	275	20	75	5	176	15	82	3	143
	LPX-AnyBURL	**79**	21	0	0	**45**	54	1	0	**63**	35	2	0	**59**	39	2	0
	LPX-AbPoint	**72**	28	0	199	**40**	58	2	523	**59**	41	0	1683	36	63	1	133
ConvE	Criage	47	53	0	0	58	42	0	0	69	30	1	0	22	78	0	0
	DP	0	99	1	0	8	75	17	0	1	97	2	0	2	93	5	0
	Imagine	0	99	1	106	54	44	2	66	2	96	2	28	2	92	6	67
	Kelpie++	0	99	1	115	18	69	13	56	1	97	2	32	13	85	2	63
	LPX-AnyBURL	**97**	3	0	0	**60**	39	1	0	**69**	30	1	0	**50**	45	5	0
	LPX-AbPoint	**77**	23	0	202	**60**	40	0	596	30	68	2	3398	**17**	81	2	279
TransE	DP	2	93	5	0	12	79	9	0	5	88	7	0	5	87	8	0
	Imagine	2	95	3	145	14	74	12	169	9	83	8	113	8	84	8	62
	Kelpie++	3	95	2	225	13	71	16	198	11	82	7	115	12	81	7	82
	LPX-AnyBURL	**66**	33	1	0	**43**	51	6	0	**65**	33	2	0	**48**	51	1	0
	LPX-AbPoint	**63**	36	1	204	**41**	58	1	604	**46**	51	3	807	**41**	56	3	242

Table 2
The explanations returned by different LPX methods for two predictions. **Type**: A = ABox, T = TBox, R = rule. **Source**: E = existing knowledge, H = hypothetical (abducted or induced) knowledge.

LPX method	Explanation	Type	Source
Prediction: Construction(a), **Dataset:** ARCO25-5			
Criage	Construction(b)	A	E
DP	Immovable(a)	A	E
Imagine	urbanRelation(a, b) and other similar assertions with a different object	A	H
Kelpie++	Immovable(a) and other related concept assertions	A	E
LPX-AnyBURL	Construction(x) ← Immovable(x)	R	H
	Immovable(a)	A	E
LPX-AbPoint	ConstructionSpace ⊑ Construction	**T**	E
	ConstructionSpace(a)	A	H
Prediction: Organization(NyenrodeUniversity), **Dataset:** YAGO4-20-C			
Criage	Organization(AGHUniversity)	A	E
DP	location(NyenrodeUniversity, Breukelen)	A	E
Imagine	location(AsburyUniversity, NyenrodeUniversity) and similar assertions with other universities as subject	A	H
Kelpie++	location(NyenrodeUniversity, Breukelen) and similar assertions with other places as object	A	E
LPX-AnyBURL	type(NyenrodeUniveristy, y) ← type(GRCollege, y)	R	H
	Organization(GRCollege)	A	E
LPX-AbPoint	School ⊑ Organization	**T**	E
	School(NyenrodeUniversity)	A	H

abduction to large-scale KGs, enabled by the generality of the approach w.r.t. the applicable reasoners. Indeed, previous abduction approaches, such as MHS-MXP, have been evaluated only on very small KGs.

Moreover, although LPX-AnyBURL provides the best overall trade-off between utility and efficiency, it requires rules that are learned from data and are not further validated. Conversely, LPX-AbPoint leverages the available schema. This matters in scenarios where explanations should remain coupled to the intended semantics established in KG construction. In this respect, we discuss the explanations reported in Tab. 2 for two predictions made via ComplEx. For the prediction on ARCO25-5, both LPX-AbPoint and LPX-AnyBURL both return useful explanations based on concept taxonomy. LPX-AbPoint uses an existing schema axiom and a hypothetical assertion, whereas the other leverages a induced rule in a different language. Differently, the hypothesis returned by Imagine seems less directly tied to the prediction. Moreover, Kelpie++ and DP show a pattern similar to the explanation from

LPX-AnyBURL, but without an explicit rule. Finally, Criage seems to make an analogy with another specific individual. For the prediction on YAGO4-20-C, the explanation by LPX-AbPoint seems more useful than the others. This highlights the differences in explanation quality that are not fully evident in the FSV distribution, which is influenced by the number of predictions for which an explanation is returned. In this case, the explanations selected by LPX-AnyBURL and Criage seem very specific because they rely on specific individuals. The assertions by Imagine, Kelpie++, and DP seem less directly related to the prediction. Moreover, Imagine seem to return a inconsistent assertion using universities rather than places in assertions on the `location` property.

6. Conclusion

We proposed LPX-AbPoint, the first LPX method that computes explanations consisting of: 1) schema axioms, 2) existing assertions, additional assertions, or both. It combines abduction and pinpointing: it supplements the KG with hypothetical assertions, and then derives an explanation from the extended KG. The abduction approach is a further generalization of the most general existing approach. For pinpointing, LPX-AbPoint integrates tracing extensions of tableau-based reasoning.

We performed a comprehensive experimental evaluation: the outcomes suggest that LPX-AbPoint outperforms approaches based purely on available knowledge and those based purely on additional knowledge and performs comparably to LPX-AnyBURL based on rule learning. We also found that LPX-AnyBURL sometimes explain predictions via very specific rules that encode similarities among individuals.

The study also suggests future research. We intend to investigate a solution assuming knowledge both at the schema and assertions level. We plan to investigate search heuristics for abduction and pinpointing based on KGE models for improving the efficiency of the LPX method.

Acknowledgments

This work was supported by (i) a Ph.D. fellowship funded within the framework of D.M. n. 630 of 24/04/2024 by the Italian MUR (*Ministero dell'Università e della Ricerca*) under the *National Recovery and Resilience Plan* (Mission 4, Component 2, Investment I.3.3) and co-funded by "Sidea Group S.r.l." for the Ph.D. project "*Semantically enriched neural-symbolic methods for Knowledge Graph Refinement and Explanation*" (CUP B91I24000240007); (ii) HPC resources and technical support from CINECA through the ISCRA initiative.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] T. Pellissier Tanon, G. Weikum, F. Suchanek, YAGO 4: A Reason-able Knowledge Base, in: A. Harth, et al. (Eds.), The Semantic Web, volume 12123, Berlin, Heidelberg, 2020, pp. 583–596. doi:10.1007/978-3-030-49461-2_34.

[2] X. L. Dong, Building a broad knowledge graph for products, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE Computer Society, Washington DC, USA, 2019, pp. 25–25. doi:10.1109/ICDE.2019.00010.

[3] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen,

J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, ACM Computing Surveys 54 (2022) 1–37. doi:10.1145/3447772.

[4] F. Baader, et al. (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, 2 ed., Cambridge University Press, 2007.

[5] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge Graph Embedding for Link Prediction: A Comparative Analysis, ACM Transactions on Knowledge Discovery from Data 15 (2021) 1–49. doi:10.1145/3424672.

[6] V. Nováček, S. K. Mohamed, Predicting polypharmacy side-effects using knowledge graph embeddings, AMIA Summits on Translational Science Proceedings 2020 (2020) 449.

[7] S. Schramm, C. Wehner, U. Schmid, Comprehensible artificial intelligence on knowledge graphs: A survey, Journal of Web Semantics 79 (2023) 100806.

[8] A. Rossi, D. Firmani, P. Merialdo, T. Teofili, Explaining Link Prediction Systems based on Knowledge Graph Embeddings, in: Z. Ives (Ed.), SIGMOD/PODS '22: Proceedings of the 2022 International Conference on Management of Data; Philadelphia, Pennsylvania, USA; 12-17 June 2022, ACM, New York, New York, USA, 2022, pp. 2062–2075. doi:10.1145/3514221.3517887.

[9] R. Peñaloza, Explaining Axiom Pinpointing, in: C. Lutz, U. Sattler, C. Tinelli, A.-Y. Turhan, F. Wolter (Eds.), Description Logic, Theory Combination, and All That, volume 11560, Springer International Publishing, Cham, 2019, pp. 475–496. doi:10.1007/978-3-030-22102-7_22.

[10] I. Tiddi, M. d'Aquin, E. Motta, An Ontology Design Pattern to Define Explanations, in: Proceedings of the 8th International Conference on Knowledge Capture, ACM, Palisades NY USA, 2015, pp. 1–8. doi:10.1145/2815833.2815844.

[11] M. Homola, J. Pukancová, J. Boborová, I. Balintová, Merge, Explain, Iterate: A Combination of MHS and MXP in an ABox Abduction Solver, in: S. Gaggl, M. V. Martinez, M. Ortiz (Eds.), Logics in Artificial Intelligence, volume 14281, Springer Nature Switzerland, Cham, 2023, pp. 338–352. doi:10.1007/978-3-031-43619-2_24.

[12] A. Steigmiller, T. Liebig, B. Glimm, Konclude: system description, Journal of Web Semantics 27 (2014) 78–85.

[13] M. Horridge, Justification based explanation in ontologies, The University of Manchester (United Kingdom), 2011.

[14] M. H. Liffiton, K. A. Sakallah, Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints, Journal of Automated Reasoning 40 (2008) 1–33. doi:10.1007/s10817-007-9084-z.

[15] I. Diliso, R. Barile, C. d'Amato, N. Fanizzi, Return of the schema: Building complete datasets for machine learning and reasoning on knowledge graphs (2026). URL: https://arxiv.org/abs/2602.14795. arXiv:2602.14795.

[16] R. Barile, C. d'Amato, N. Fanizzi, LP-DIXIT: Evaluating explanations for link predictions on knowledge graphs using large language models, in: Proceedings of the ACM on Web Conference 2025, WWW '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 4034–4042. URL: https://doi.org/10.1145/3696410.3714667. doi:10.1145/3696410.3714667.

[17] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models (2023). arXiv:2303.18223.

[18] H. Zhang, T. Zheng, J. Gao, C. Miao, L. Su, Y. Li, K. Ren, Data Poisoning Attack against Knowledge Graph Embedding, in: S. Kraus (Ed.), IJCAI '19: Proceedings of the 28th International Joint Conference on Artificial Intelligence; Macao, China; 10-16 August 2019, IJCAI, Online, 2019, pp. 4853–4859. doi:10.24963/ijcai.2019/674.

[19] P. Pezeshkpour, C. A. Irvine, Y. Tian, S. Singh, Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications, in: Burstein, Jill, Doran, Christy, Solorio, Thamar (Eds.), NAACL-HLT '19: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); Minneapolis, Minnesota, USA; 02-07 June 2019, volume 1, Association for Computational Linguistics, Kerrville, Texas, USA, 2019, pp. 3336–3347. doi:10.18653/v1/N19-1337.

[20] R. Barile, C. d'Amato, N. Fanizzi, Explanation of Link Predictions on Knowledge Graphs via

Levelwise Filtering and Graph Summarization, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), Proceedings of the 26th European Semantic Web Conference (ESWC 2024), volume 14664, Springer Nature Switzerland, Cham, 2024, pp. 180–198. doi:10.1007/978-3-031-60626-7_10.

[21] D. Zhao, G. Wan, Y. Zhan, Z. Wang, L. Ding, Z. Zheng, B. Du, KE-X: Towards subgraph explanations of knowledge graph embedding based on knowledge information gain, Knowledge-Based Systems 278 (2023) 110772. doi:10.1016/j.knosys.2023.110772.

[22] T. Ma, X. Song, W. Tao, M. Li, J. Zhang, X. Pan, J. Lin, B. Song, X. Zeng, KGExplainer: Towards Exploring Connected Subgraph Explanations for Knowledge Graph Completion (2024). arXiv:2404.03893.

[23] V. Baltatzis, L. Costabello, Kgex: explaining knowledge graph embeddings via subgraph sampling and knowledge distillation, in: Learning on Graphs Conference, PMLR, 2024, pp. 27–1.

[24] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, H. Chen, S. Culpepper, Interaction Embeddings for Prediction and Explanation in Knowledge Graphs, in: WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, February 11-15, 2019, ACM, New York, New York, USA, 2019, pp. 96–104. doi:10.1145/3289600.3291014.

[25] C. d'Amato, P. Masella, N. Fanizzi, An Approach Based on Semantic Similarity to Explaining Link Predictions on Knowledge Graphs, in: J. He, R. Unland, E. J. Santos, X. Tao, H. Purohit, W.-J. van den Heuvel, J. Yearwood, J. Cao (Eds.), IEEE/WIC/ACM International Conference on Web Intelligence, ACM, New York, New York, USA, 2021, pp. 170–177. doi:10.1145/3486622.349395.

[26] E. Amador-Domínguez, E. Serrano, D. Manrique, GEnI: A framework for the generation of explanations and insights of knowledge graph embedding predictions, Neurocomputing 521 (2023) 199–212. doi:10.1016/j.neucom.2022.12.010.

[27] P. Betz, C. Meilicke, H. Stuckenschmidt, Adversarial Explanations for Knowledge Graph Embeddings, in: L. De Raedt (Ed.), IJCAI '22: Proceedings of the 31th International Joint Conference on Artificial Intelligence; Vienna, Austria; 23-29 July 2022, IJCAI, Online, 2022, pp. 2820–2826. doi:10.24963/ijcai.2022/391.

[28] N. A. Krishnan, C. R. Rivero, A Model-Agnostic Method to Interpret Link Prediction Evaluation of Knowledge Graph Embeddings, in: I. Frommholz (Ed.), CIKM'23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, ACM, 2023, pp. 1107–1116. doi:10.1145/3583780.3614763.

[29] Y. Ismaeil, D. Stepanova, T.-K. Tran, H. Blockeel, FeaBI: A Feature Selection-Based Framework for Interpreting KG Embeddings, in: T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I, Springer, Cham, Switzerland, 2023, pp. 599–617. doi:10.1007/978-3-031-47240-4_32.

[30] B. Steenwinckel, G. Vandewiele, M. Weyns, T. Agozzino, F. D. Turck, F. Ongenae, INK: Knowledge graph embeddings for node classification, Data Mining and Knowledge Discovery 36 (2022) 620–667. doi:10.1007/s10618-021-00806-z.

[31] J. Lajus, L. Galárraga, F. Suchanek, Fast and exact rule mining with amie 3, in: The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17, Springer, 2020, pp. 36–52.

[32] C. Meilicke, M. W. Chekol, P. Betz, M. Fink, H. Stuckeschmidt, Anytime bottom-up rule learning for large-scale knowledge graph completion, The VLDB Journal 33 (2024) 131–161.

[33] D. Calvanese, M. Ortiz, M. Simkus, G. Stefanoni, Reasoning about explanations for negative query answers in DL-Lite, Journal of Artificial Intelligence Research 48 (2013) 635–669.

[34] I. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavicius, Explanations for negative query answers under existential rules (2020).

[35] F. Wei-Kleiner, Z. Dragisic, P. Lambrix, Abduction framework for repairing incomplete EL ontologies: Complexity results and algorithms, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 28, 2014.

[36] C. Elsenbroich, O. Kutz, U. Sattler, A case for abductive reasoning over ontologies, in: Proceedings

of the OWLED 2006 Workshop on OWL: Experiences and Directions, Athens, Georgia, USA, November 10-11, 2006, volume 216, CEUR, 2006, pp. 1–12.
[37] M. Bienvenu, Complexity of Abduction in the EL Family of Lightweight Description Logics., in: KR, 2008, pp. 220–230.
[38] R. Reiter, A theory of diagnosis from first principles, Artificial intelligence 32 (1987) 57–95.
[39] K. M. Shchekotykhin, D. Jannach, T. Schmitz, MergeXplain: Fast Computation of Multiple Conflicts for Diagnosis., in: IJCAI, volume 15, 2015, pp. 3221–3228.
[40] U. Junker, Quickxplain: Preferred explanations and relaxations for over-constrained problems, in: Proceedings of the 19th National Conference on Artifical Intelligence, 2004, pp. 167–172.
[41] P. Rodler, Understanding the QuickXPlain Algorithm: Simple Explanation and Formal Proof (2022). doi:10.48550/arXiv.2001.01835. arXiv:2001.01835.
[42] I. Mozetič, C. Holzbaur, Controlling the complexity in model-based diagnosis, Annals of Mathematics and Artificial Intelligence 11 (1994) 297–314. doi:10.1007/BF01530747.
[43] R. R. Hoffman, S. T. Mueller, G. Klein, J. Litman, Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance, Frontiers in Computer Science 5 (2023) 1096257. doi:10.3389/fcomp.2023.1096257.
[44] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, pp. 2787–2795.
[45] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11573. doi:10/gqjp9q, issue: 1.
[46] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International Conference on Machine Learning, JMLR, Online, 2016, pp. 2071–2080. doi:10.5555/3045390.3045609.
[47] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical owl-dl reasoner, Journal of Web Semantics 5 (2007) 51–53.
[48] R. Barile, C. d'Amato, N. Fanizzi, GRainsaCK: a comprehensive software library for benchmarking explanations of link prediction tasks on knowledge graphs, arXiv preprint arXiv:2508.08815 (2025).
[49] P. Betz, L. Galárraga, S. Ott, C. Meilicke, F. Suchanek, H. Stuckenschmidt, Pyclause-simple and efficient rule handling for knowledge graphs, in: Thirty-Third International Joint Conference on Artificial Intelligence {IJCAI-24}, International Joint Conferences on Artificial Intelligence Organization, 2023, pp. 8610–8613.