Enabling Abductive Learning to Exploit Knowledge Graph

Yu-Xuan Huang, **Zequn Sun**, **Guangyao Li**, **Xiaobin Tian**,
Wang-Zhou Dai, **Wei Hu**, **Yuan Jiang** and **Zhi-Hua Zhou**
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{huangyx, daiwz, jiangy, zhouzh}@lamda.nju.edu.cn,
{zqsun, gyli, xbtian}.nju@gmail.com, whu@nju.edu.cn

Abstract

Most systems integrating data-driven machine learning with knowledge-driven reasoning usually rely on a specifically designed knowledge base to enable efficient symbolic inference. However, it could be cumbersome for the nonexpert end-users to prepare such a knowledge base in real tasks. Recent years have witnessed the success of large-scale knowledge graphs, which could be ideal domain knowledge resources for real-world machine learning tasks. However, these large-scale knowledge graphs usually contain much information that is irrelevant to a specific learning task. Moreover, they often contain a certain degree of noise. Existing methods can hardly make use of them because the large-scale probabilistic logical inference is usually intractable. To address these problems, we present ABductive Learning with Knowledge Graph (ABL-KG) that can automatically mine logic rules from knowledge graphs during learning, using a knowledge forgetting mechanism for filtering out irrelevant information. Meanwhile, these rules can form a logic program that enables efficient joint optimization of the machine learning model and logic inference within the Abductive Learning (ABL) framework. Experiments on four different tasks show that ABL-KG can automatically extract useful rules from large-scale and noisy knowledge graphs, and significantly improve the performance of machine learning with only a handful of labeled data.

1 Introduction

The integration of machine learning and logical reasoning is a long-standing holy grail problem of Artificial Intelligence (AI). It is also argued that the next generation of AI calls for integrating the power of machine learning and logical reasoning [Bengio, 2017]. In recent years, Neuro-Symbolic (NeSy) Learning [Garcez *et al.*, 2019; Raedt *et al.*, 2020] and Statistical Relational AI (StarAI) [Raedt *et al.*, 2016] are representative progress. NeSy systems usually aim to build an explainable neural network structure with external domain knowledge. StarAI [Koller *et al.*, 2007] shares a similar idea, where a probabilistic graphical model is constructed based on domain knowledge expressed in first-order logic (FOL). Probabilistic Logic Program (PLP) [De Raedt and Kimmig, 2015] combines these two paradigms by extending FOL to accommodate probabilistic groundings such that probabilistic inference can be conducted.

Abductive Learning (ABL) [Zhou, 2019; Zhou and Huang, 2022] is a flexible framework that integrates a machine learning model with a FOL reasoning model while preserving the full expressive power of both sides: the machine learning model learns to convert input data into primitive logic facts, named pseudo-labels, which serve as input to symbolic reasoning; the reasoning model tries to infer the truth-value of these facts to update the machine learning model. The integration of the two systems adopts abduction, a.k.a. abductive reasoning, to reason about the pseudo-labels based on background FOL knowledge in the reasoning model.

In order to exploit logical reasoning in machine learning, most of the above systems assume a specifically-designed background knowledge base (KB) containing FOL rules for the learning task, which could be unrealistic in real-world tasks. Although recent progress on ABL [Dai and Muggleton, 2021] has shown its capability of learning with incomplete background knowledge, the knowledge bases are still manually prepared by human experts. A specific expert knowledge base would undoubtedly benefit machine learning tasks, however, it could be hard to obtain in reality due to the following reasons: (1) Domain expertise may be very expensive and developing a large knowledge base consisting of abundant and correct rules is time-consuming. (2) These knowledge bases are highly task-specific and can hardly be reused in different tasks.

Recently, knowledge graphs (KGs), as a structured form of human knowledge, have drawn great attention from both academia and industry [Nickel *et al.*, 2015; Wang *et al.*, 2017; Ji *et al.*, 2022; Hogan *et al.*, 2022]. KGs model information in the form of a graph, consisting of entities (nodes) and relations between them (edges). A large number of KGs have been created, including YAGO [Rebele *et al.*, 2016], DBpedia [Lehmann *et al.*, 2015], NELL [Carlson *et al.*, 2010], Freebase [Bollacker *et al.*, 2008], ConceptNet [Speer *et al.*, 2017], etc. They contain millions of nodes and billions of edges [Nickel *et al.*, 2015]. If existing KGs could be used as resources for domain knowledge, from which we extract a segment of knowledge related to the machine learning task, the problem of lacking knowledge can be alleviated to a certain extent. Note that the words *knowledge base* (KB) and *knowledge graph* (KG) may share the same meaning in some papers. In this paper, we distinguish between KB and KG, where KG contains triplets and KB consists of FOL rules.

Three challenges hinder machine learning systems from exploiting knowledge in KGs. First, large-scale and general-purpose KGs often contain much information irrelevant to the machine learning task, causing inefficiency or inconsistency in reasoning. Second, a KG that is not tailored for the learning task may use different expressions for entities and relations, bringing semantic challenges in entity matching, e.g., "father" in the KG but "dad" in the machine learning task. Third, large-scale KGs are usually mined from data, the noise within could bring obstacles to rule extraction and logical inference.

This paper proposes to tackle the above challenges under the abductive learning (ABL) framework and presents an approach to facilitate ABL systems with knowledge graphs. Given a learning task with a large-scale and general-purpose KG, the proposed ABL-KG (ABductive Learning with Knowledge Graph) first mines logic rules from a knowledge sub-graph relevant to the learning task. Then, it adapts the rules for the learning task by aligning their names. We propose a remembering algorithm to extract and transform the mined rules to a weighted logic program while resolving the contradictions caused by KG noise. Finally, ABL-KG leverages the logic program as background knowledge in ABL by utilizing a consistency measure to handle noisy and uncertain FOL rules.

We assess the effectiveness and generalization of ABL-KG on a diverse range of tasks, including a classification task on tabular data, two representation learning tasks on graph data, and a classification task on image data. Experiments show that ABL-KG can exploit the knowledge in KGs without manual efforts, and improve machine learning performance by automatically extracting task-relevant domain knowledge from large-scale and general-purpose KGs.

2 Related Work

Probabilistic Logic Program (PLP) [De Raedt and Kimmig, 2015] and Statistical Relational Learning (SRL) [Koller *et al.*, 2007; Raedt *et al.*, 2016] are two typical paradigms for integrating logical reasoning and machine learning. PLP extends FOL to accommodate probabilistic groundings and conduct probabilistic inference. SRL tries to construct a probabilistic graphical model based on domain knowledge expressed in FOL. Various novel approaches have been proposed in recent years, including DeepProbLog [Manhaeve *et al.*, 2018], Abductive Learning [Zhou, 2019] and NGS [Li *et al.*, 2020].

A knowledge graph (KG) has a multi-relational graph structure, with each node representing an entity and each edge indicating the relation between two connected entities. Such structured commonsense or factual knowledge has been widely used in intelligent applications [Ji *et al.*, 2022]. There are two typical approaches to exploiting KGs. The conventional symbolic methods either mine logic rules to infer new facts [Galárraga *et al.*, 2013; Galárraga *et al.*, 2015; d'Amato *et al.*, 2016], or use keyword or SPARQL queries executed on a KG to retrieve answers [Unger *et al.*, 2012]. Recently, representation learning can capture KG semantics in vector space and use the acquired embeddings to improve downstream tasks such as link prediction [Wang *et al.*, 2017] and entity alignment [Sun *et al.*, 2020].

Some approaches attempt to combine deep neural networks with KGs [Marino *et al.*, 2017; Wang *et al.*, 2018] by building an end-to-end graph neural network (GNN), which often demand a large number of labeled data for training and the expressiveness of knowledge is limited. Instead of using GNN, Scallop [Huang *et al.*, 2021a] scales up DeepProblog by using approximate probabilistic logic inference, and it can use small-scale KGs as knowledge resources. Different from these approaches, ABL-KG could leverage reasonably large-scale KGs with unlabeled data by abductive learning and preserve the expressiveness of FOL at the same time.

The ability to discard irrelevant information is a key feature for an intelligent agent, which is referred to as forgetting [Lin and Reiter, 1994]. Forgetting preserves all logical consequences of relevant symbols while removing the irrelevant ones, which enables a logic program to adequately and efficiently handle reasoning tasks such as query answering. Various forgetting operations [Lin and Reiter, 1994; Wang *et al.*, 2005; Delgrande, 2014] have been proposed for different types of logic programs. For a comprehensive overview of forgetting, please refer to [Eiter and Kern-Isberner, 2019]. While these works usually focus on forgetting a small subset of irrelevant information in a logic program, our work would discard a large proportion of them efficiently.

3 Preliminaries

3.1 Abductive Learning

Abductive Learning (ABL) [Zhou, 2019; Zhou and Huang, 2022] is a framework that integrates a machine learning model and a logical reasoning model. The machine learning model f maps the unlabeled input data $\boldsymbol{x} \in \boldsymbol{X}_u$ into discrete symbols $\boldsymbol{y} \in \mathcal{Y}$, which are called *pseudo-labels* since no supervision on $\boldsymbol{y}$. The reasoning model receives the symbols $\boldsymbol{y}$ and verifies their consistency with a knowledge base $\mathcal{KB}$ of FOL rules. If inconsistent, the reasoning model would correct the pseudo-labels $\boldsymbol{y}$ to abduced labels $\bar{\boldsymbol{y}}$ by abductive reasoning. For instance, given a crayfish's features $\boldsymbol{x}$, an under-trained classifier f may output the pseudo-label $\boldsymbol{y} = fish$, then the reasoning model is expected to identify the inconsistency and revise it to $\bar{\boldsymbol{y}} = crustacean$.

Abductive reasoning (abduction) is a form of logical inference that seeks grounded facts explaining observations based on background knowledge. To illustrate clearly, this paper denotes logical symbols as follows: "$\wedge$" is conjunction (and); "$\vee$" is disjunction (or); "$\leftarrow$" is implication, which means that if premises on the right of "$\leftarrow$" hold, then the conclusion on the left holds. For example, consider a $\mathcal{KB}$ containing the rules:

$$backbone(X) \leftarrow fish(X), \tag{1}$$

$$\neg backbone(X) \leftarrow insect(X) \vee crustacean(X), \tag{2}$$

$$false \leftarrow insect(X) \wedge crustacean(X), \tag{3}$$

where the first two rules describe the characteristics of $fish$, $insect$ and $crustacean$, and the last rule specifies that an animal cannot be both an $insect$ and a $crustacean$ simultaneously. Given the observation that an animal (e.g., a crayfish)

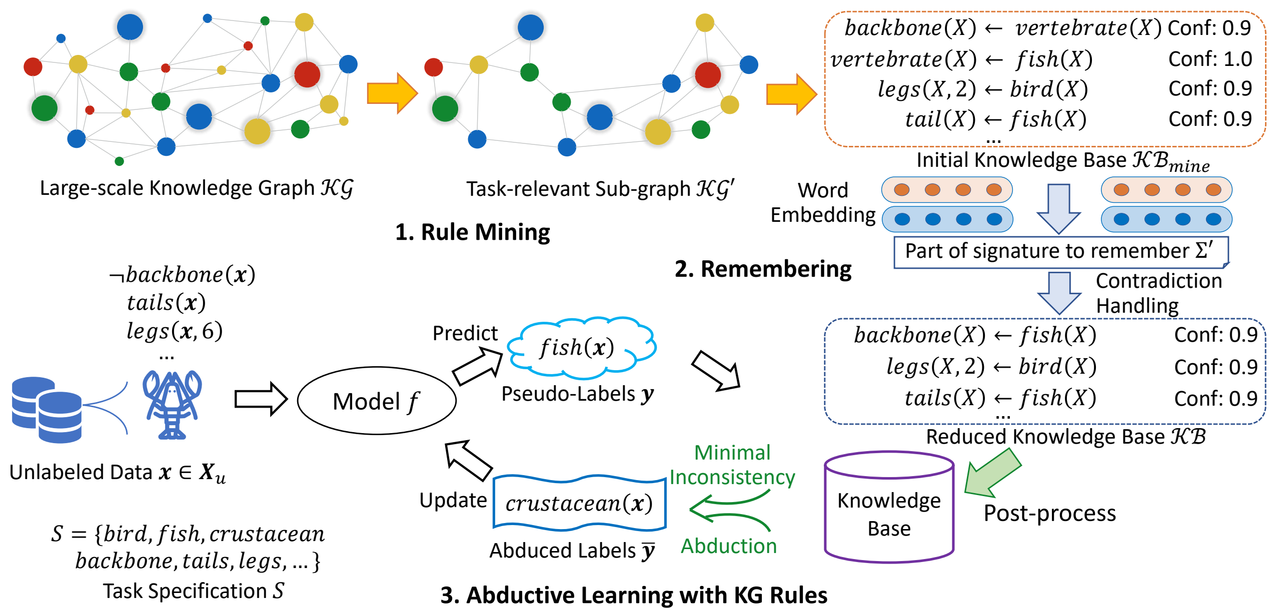

Figure 1: The ABL-KG framework. The first step involves mining rules that are relevant to the machine learning task from the knowledge graph. Then, it conducts a remembering procedure to generate rules containing the target predicates without loss of other information. Finally, the rules are utilized for abductive learning with a newly designed consistency measure for noisy and uncertain rules.

has no backbone, Rule (1) implies that it should not be a $fish$ (inconsistent with the pseudo-label). Continuing this reasoning process, based on Rules (2) and (3), both *insect* and $crustacean$ could be two possible explanations. If other rules in $\mathcal{KB}$ indicate that it is not an $insect$, then a $crustacean$ would be the only explanation (abduced label).

Usually, there are multiple candidate abduced labels (denoted by $\bar{\mathcal{Y}}$), and ABL minimizes inconsistency to choose the best $\bar{\boldsymbol{y}} \in \bar{\mathcal{Y}}$ based on a *consistency measure*. After revising $\boldsymbol{y}$ to $\bar{\boldsymbol{y}}$, ABL treats them as ground-truth labels to update the machine learning model f. The above process repeats iteratively and f improves its performance by performing abduction on the unlabeled data $\boldsymbol{X}_u$ with knowledge base $\mathcal{KB}$.

3.2 Knowledge Graph

In this work, we focus on the RDF (Resource Description Framework) KGs [Schreiber and Raimond, 2014]. It is a structured representation of facts regarding entities and relations. Each fact is a triplet of the form $(h, \mathtt{r}, t)$, e.g., $(bird, \mathtt{HasA}, feather)$. Subject h and object t are entities, which can be real-world instances or concepts, and $\mathtt{r}$ is their relation. In predicate logic, $(h, \mathtt{r}, t)$ is equivalent to $\mathtt{r}(h, t)$.

3.3 Forgetting and Remembering

Forgetting, originally proposed in [Lin and Reiter, 1994], aims at removing irrelevant information from a knowledge base $\mathcal{KB}$. In this paper, we denote the signature (i.e., all the predicates) of $\mathcal{KB}$ by Σ, and $\Sigma' \subseteq \Sigma$ is the part to forget. $\mathrm{forget}(\mathcal{KB}, \Sigma')$ is the result of forgetting Σ' from $\mathcal{KB}$. The dual of forgetting is remembering [Lin and Reiter, 1994], denoted by $\mathrm{remember}(\mathcal{KB}, \Sigma') = \mathrm{forget}(\mathcal{KB}, \Sigma \setminus \Sigma')$, which means forgetting the remaining signature while preserving all logical consequences in $\mathcal{KB}$ w.r.t. Σ'.

For instance, assume that $\mathcal{KB}$ consists of two rules $backbone(X) \leftarrow vertebrate(X)$ and $vertebrate(X) \leftarrow fish(X)$. The predicate "$vertebrate$" is irrelevant to the learning task and we want to forget it. In this case, signature $\Sigma = \{fish, vertebrate, backbone\}$ and $\Sigma' = \{vertebrate\}$, and the result of $\mathrm{forget}(\mathcal{KB}, \Sigma')$ is $\{backbone(X) \leftarrow fish(X)\}$. Obviously, forgetting $vertebrate$ not only keeps information of the remaining signature, but also leads to a new knowledge base that could perform reasoning more efficiently.

4 The ABL-KG Framework

4.1 Problem Setting

We are given unlabeled data $\boldsymbol{X}_u$, a large-scale knowledge graph $\mathcal{KG}$, and an initial machine learning model f. To utilize the semantics of data, a task specification S of $\boldsymbol{X}_u$ is available, e.g., the set of class names and attribute names such as $bird$ or $backbone$ in a classification task. The goal is to leverage $\boldsymbol{X}_u$ and $\mathcal{KG}$ to improve the performance of f.

4.2 Framework

The ABL-KG framework can be roughly divided into three main steps: rule mining, remembering, and abductive learning with KG rules. Figure 1 and Algorithm 1 show its outline.

Rule Mining. Given a large-scale knowledge graph $\mathcal{KG}$ and a machine learning task, $\mathcal{KG}$ contains numerous irrelevant information to the task. Therefore, ABL-KG first extracts a sub-graph $\mathcal{KG}' \subset \mathcal{KG}$ relevant to the machine learning task, where the number of triplets $|\mathcal{KG}'| \ll |\mathcal{KG}|$. The relevance is measured by the semantics of the relations and entities in $\mathcal{KG}$ and S. Then, a rule mining algorithm is employed to mine FOL rules from the sub-graph $\mathcal{KG}'$, which forms the initial knowledge base $\mathcal{KB}_{mine}$. The upper part of Figure 1 shows an illustration, further details are introduced in Section 4.3.

Algorithm 1 Abductive Learning with Knowledge Graph

Input: Knowledge graph $\mathcal{KG}$; Unlabeled data $\boldsymbol{X}_u$; Task specification S; Initial model f; Labeled data $\boldsymbol{X}_l$
Output: Machine learning model f

1: $\mathcal{KG}' \leftarrow \mathrm{SubGraph}(\mathcal{KG}, S, d)$ // Sub-graph depth d
2: $\mathcal{KB}_{mine} \leftarrow \mathrm{MineRules}(\mathcal{KG}')$
3: $\Sigma' \leftarrow \mathrm{SigMatch}(\mathcal{KB}_{mine}, S)$
4: $\mathcal{KB} \leftarrow \mathrm{Remember}(\mathcal{KB}_{mine}, \Sigma')$
5: $\mathcal{KB} \leftarrow \mathrm{PostProcess}(\mathcal{KB}, \boldsymbol{X}_l)$
6: **for** $t = 1$ **to** T **do**
7: $\quad \boldsymbol{Y}_u \leftarrow f(\boldsymbol{X}_u)$
8: $\quad \bar{\boldsymbol{Y}}_u \leftarrow \mathrm{Abduce}(\boldsymbol{Y}_u, \boldsymbol{X}_u, \mathcal{KB})$ // Solve Eq. (4)
9: $\quad f \leftarrow \mathrm{Update}(f, \boldsymbol{X}_u, \bar{\boldsymbol{Y}}_u)$ // Update f with $\bar{\boldsymbol{Y}}_u$
10: **end for**

Remembering. To preserve the connectivity of the extracted sub-graph, the FOL rule set $\mathcal{KB}_{mine}$ may still contain some predicates not matched to the names in the task specification S, and hence, it could further be reduced. As shown in the top right of Figure 1, the predicates $backbone$, $fish$, $legs$, $bird$ and $tails$ should be directly mapped to S, while the predicate $vertebrate$ has no matched words in S and can be removed. Thus, we propose a *remembering* algorithm to do this. The new $\mathcal{KB}$ only consists of the target predicates (signature) Σ' without the loss of other information, e.g., keeping the relation between predicates $fish$ and $backbone$ as shown on the right side of Figure 1. In addition, since $\mathcal{KG}$ inevitably contains noise, we design a strategy to handle contradictory rules in $\mathcal{KB}$ during the remembering process. More details are introduced in Section 4.4.

Abductive Learning with KG Rules. The lower part of Figure 1 and Lines 6–10 in Algorithm 1 show the basic procedures in abductive learning (ABL). Here a key challenge is how to utilize the noisy $\mathcal{KB}$ in ABL. If a small amount of labeled data is available, ABL-KG can use it to post-process and filter unreliable rules in $\mathcal{KB}$. Besides, in abductive learning, one needs to minimize the inconsistency between abduced labels and $\mathcal{KB}$. Due to the inherent noise in $\mathcal{KB}$, a fully consistent abduced label may not even exist. Therefore, we propose to formulate $\mathcal{KB}$ as a weighted logic program and introduce a new consistency measure to handle the uncertainty in the abductive reasoning process. Details can be found in Section 4.5.

4.3 Rule Mining

Sub-graph Extraction

Sub-graph extraction aims at extracting a sub-graph $\mathcal{KG}'$ of $\mathcal{KG}$ relevant to the given machine learning task. Let $\mathcal{R} = \pi_{relation}(\mathcal{KG})$ denote the set of relations in $\mathcal{KG}$ and $\mathcal{E} = \pi_{entity}(\mathcal{KG})$ the set of entities. ABL-KG first finds a set $ER \subseteq \mathcal{R} \cup \mathcal{E}$ of entities and relations in $\mathcal{KG}$ that have the same names as elements in task specification S. Then, a sub-graph $\mathcal{KG}' = \{(h, r, t) \in \mathcal{KG} \mid \{h, r, t\} \cap ER \neq \emptyset\}$ consists of the triplets that contain the relations or entities in ER. Furthermore, we could construct a larger $\mathcal{KG}'$ based on current $\mathcal{KG}'$ by iteratively performing the above procedure. ABL-KG limits the size of the sub-graph $\mathcal{KG}'$ by a parameter d, which controls the depth of graph expansion.

Rule Mining for Knowledge Graph

ABL-KG could use any rule mining algorithms designed for KG to mine association rules from the extracted sub-graph $\mathcal{KG}'$. The outputs (knowledge base $\mathcal{KB}_{mine}$) are typically weighted first-order Horn rules. Each rule's literals are triplets with variables, e.g., $isCitizenOf(C, S) \leftarrow hasChild(P, C) \wedge isCitizenOf(P, S)$ with confidence 0.9 estimated from data provided by the rule mining algorithm.

We also propose a simple strategy that directly converts triplets in $\mathcal{KG}'$ into rules. We first define a subset of relations in $\mathcal{KG}'$ related to the learning task, and the corresponding direction of implication and confidence in logic rules. Then, the entities are directly converted into predicates with the same name. If a number exists in an entity, it serves as the second argument in its corresponding atom. For example, the triplets $(fish, IsA, vertebrate)$ and $(bird, HasA, two\_legs)$ can be transformed into rules $vertebrate(X) \leftarrow fish(X)$ and $legs(X, 2) \leftarrow bird(X)$, respectively. The confidence of the mined rules is generated by a user-defined tolerance w.r.t. the relation in the triplets to reflect the estimated noise level in $\mathcal{KG}'$, and details are provided in the experiment section.

4.4 Remembering

Signature Matching

Let Σ be the signature in the mined knowledge base $\mathcal{KB}_{mine}$. Then, the target signature (signature to remember) is $\Sigma' = \{\sigma \in \Sigma \mid \mathrm{MaxSimilarity}(\sigma, S) > \tau\}$, where $\mathrm{MaxSimilarity}()$ returns the maximum similarity between predicate σ and the elements in the task specification S, estimated from word embeddings, and τ is a threshold.

Remembering Algorithm

We propose a remembering algorithm that produces a new knowledge base $\mathcal{KB}$ from $\mathcal{KB}_{mine}$ that only contains the target signature Σ' without any loss of other information. The signature to remember in ABL-KG is often much smaller than the entire signature of $\mathcal{KB}_{mine}$, i.e., $|\Sigma'| \ll |\Sigma|$.

Algorithm 2 shows an outline. The algorithm starts from a subset R_{new} of input $\mathcal{KB}_{mine}$, where the signature $\mathrm{Sig}(r)$ of each rule r contains at least one element of the target signature Σ'. In the outer loop (cf. Lines 3–16), it derives a set of new clauses R_{res} by resolving every clause in R_{new} and every clause in $\mathcal{KB}$ w.r.t. the signature to forget ($\Sigma \setminus \Sigma'$), i.e., $\mathrm{ResolutionOn}(r_{new}, r, \Sigma \setminus \Sigma') = \{\alpha \vee \beta \mid r_{new} = \alpha \vee p, r = \beta \vee \neg p, p \in \Sigma \setminus \Sigma'\}$. If resolution succeeds and the resolvent is not in R_{res}, it is added to R_{res} and the current $\mathcal{KB}$. The resolvent's confidence would be set as the product of the confidence of the two resolved clauses. In the next iteration, the set of new resolvents R_{res} will be used as R_{new} and resolves with the current $\mathcal{KB}$. The above routine conducts reasoning related to Σ' on $\mathcal{KB}$ until saturated, i.e., there exists no "hidden" information about Σ' that is entailed by $\mathcal{KB}$ but not directly asserted in it. Finally, a new knowledge base is returned by retrieving the clauses in the current $\mathcal{KB}$ that only contain Σ'.

The basic idea of our remembering algorithm is to avoid unnecessary resolutions by restricting one of the clauses in resolution to contain target signatures. For example, given a knowledge base $\mathcal{KB}_{mine} = \{a \leftarrow b, b \leftarrow c, d \leftarrow e, e \leftarrow d\}$, we want to remember literals a and c, and the result is $\{a \leftarrow c\}$.

Algorithm 2 Remembering

Input: Knowledge base $\mathcal{KB}_{mine}$; Entire signature Σ; Target signature Σ'
Parameter: Iteration limit T
Output: Reduced knowledge base $\mathcal{KB}$

$R_{new} \leftarrow \{r \in \mathcal{KB}_{mine} \mid \mathrm{Sig}(r) \cap \Sigma' \neq \emptyset\}$
$\mathcal{KB} \leftarrow \mathcal{KB}_{mine}$
while $t \leq T$ and $R_{new} \neq \emptyset$ **do**
  $R_{res} \leftarrow \emptyset$
  for $r_{new} \in R_{new}$ **do**
    for $r \in \mathcal{KB}$ **do**
      $r_{res} \leftarrow \mathrm{ResolutionOn}(r_{new}, r, \Sigma \setminus \Sigma')$
      if $r_{res} \neq \emptyset$ and $r_{res} \notin \mathcal{KB}$ **then**
        $R_{res} \leftarrow R_{res} \cup \{r_{res}\}$
      end if
    end for
  end for
  $R_{new} \leftarrow R_{res}$
  $\mathcal{KB} \leftarrow \mathcal{KB} \cup R_{res}$
  $t \leftarrow t + 1$
end while
$\mathcal{KB} \leftarrow \{r \in \mathcal{KB} \mid Sig(r) \subseteq \Sigma'\}$

If resolving on b, d, e succeeds, the resolution on d and e would be unnecessary because they are completely irrelevant to a and c. In short, our algorithm improves the efficiency of forgetting. Moreover, even if irrelevant symbols are unsatisfiable, e.g., $\mathcal{KB}_{mine}$ becomes $\{a \leftarrow b, b \leftarrow c, d, \neg d\}$, it would not influence the ABL-KG framework, since we only care about the relations within the target signature $\Sigma' = \{a, c\}$.

Theorem 1. *Let $\mathcal{KB}_{mine}$ be any finite consistent propositional or grounded first-order answer set program [Lifschitz, 2002] consisting of Horn clauses, if the iteration limit T is sufficiently large, Algorithm 2 always terminates and returns a result of forgetting about $\Sigma \setminus \Sigma'$ in $\mathcal{KB}_{mine}$.*

We defer the proof to Appendix. Theorem 1 shows that the algorithm can output a correct logic program of forgetting.

Contradiction Handling

The rules mined from $\mathcal{KG}'$ may contain contradictory rules. For example, $animal(X) \leftarrow bird(X)$ and $\neg animal(X) \leftarrow bird(X)$ may both exist in $\mathcal{KB}$ (due to $(bird, \texttt{IsA}, animal)$ and $(bird, \texttt{Antonym}, animal)$ in ConceptNet [Speer *et al.*, 2017]). They would result in an incorrect rule $\neg bird(X)$ after remembering, which means that everything is not a bird. We define contradictory rules as follows: if both $p \leftarrow \alpha$ and $\neg p \leftarrow \alpha$ exist in $\mathcal{KB}$, then they are contradictory rules. ABL-KG detects contradictions during the remembering process. If r_1 and r_2 are contradictory rules, it removes r_1, r_2 and their resolvents from the current $\mathcal{KB}$.

4.5 Abductive Learning with KG Rules

Rule Post-processing

If a small amount of labeled data are available, they could be used not only for training the initial machine learning model f, but also for rule post-processing. By evaluating the quality of rules, ABL-KG drops low-quality rules and adjusts the rules' confidence to $\beta \cdot conf_{rule} + (1-\beta) \cdot conf_{data}$, where $conf_{rule}$ is the rule's confidence given by $\mathcal{KG}$, and $conf_{data}$ given by labeled data.

Consistency Measure for Noisy and Uncertain Rules

Previous ABL methods adopt a consistency measure for abduction to find the best abduced labels, which usually requires that the abduced labels should be consistent with $\mathcal{KB}$, i.e., all the rules in $\mathcal{KB}$ are satisfied. However, faced with inevitably noisy $\mathcal{KG}$, even though the mined rules are post-processed, they would not always be satisfied (unless all of them have confidence 1.0), in which case the abduced labels that are consistent with $\mathcal{KB}$ may not even exist. Therefore, we propose a new consistency measure for noisy and uncertain rules, where the optimization problem can be formalized as:

$$\max_{\bar{\boldsymbol{y}} \in \bar{\mathcal{Y}}} \mathrm{ModelScore}(f, \boldsymbol{x}, \bar{\boldsymbol{y}}) + \alpha \cdot \mathrm{KBScore}(\mathcal{KB}, \boldsymbol{x}, \bar{\boldsymbol{y}}), \quad (4)$$

where $\bar{\mathcal{Y}}$ is the set of candidate abduced labels, and $\boldsymbol{x}$ and $\bar{\boldsymbol{y}}$ are unlabeled data and abduced labels, respectively.

$\mathrm{ModelScore}(f, \boldsymbol{x}, \bar{\boldsymbol{y}})$ represents the consistency between abduced labels and model. For example, it can be defined as:

$$\mathrm{ModelScore}(f, \boldsymbol{x}, \bar{\boldsymbol{y}}) = \sum_{x_i \in \boldsymbol{x}} \mathrm{Conf}(x_i, \bar{y}_i), \quad (5)$$

where $\mathrm{Conf}(x_i, \bar{y}_i)$ denotes the confidence by model f that sample x_i belongs to label $\bar{y}_i$. Alternatively, other consistency measures, e.g., [Huang *et al.*, 2020; Cai *et al.*, 2021; Huang *et al.*, 2021b; Huang *et al.*, 2023], could also be used.

$\mathrm{KBScore}(\mathcal{KB}, \boldsymbol{x}, \bar{\boldsymbol{y}})$ represents the consistency between abduced labels and knowledge base $\mathcal{KB}$, which is defined as:

$$\mathrm{KBScore}(\mathcal{KB}, \boldsymbol{x}, \bar{\boldsymbol{y}}) = -\sum_{r \in \mathrm{InconsRules}(\mathcal{KB}, \boldsymbol{x}, \bar{\boldsymbol{y}})} \mathrm{Weight}(r), \quad (6)$$

where r is a ground rule in $\mathcal{KB}$ which is inconsistent with $\boldsymbol{x}, \bar{\boldsymbol{y}}$, i.e., $r \cup \boldsymbol{x} \cup \bar{\boldsymbol{y}}$ is inconsistent. A ground rule is a rule where all variables are replaced by specific instances. Take the example in Figure 1, if abduced label $\bar{\boldsymbol{y}}$ is $fish$, the ground rule $backbone(\boldsymbol{x}) \leftarrow fish(\boldsymbol{x})$ is violated and therefore inconsistent with abduced labels. $\mathrm{Weight}(r)$ is the weight of rule r, which can be calculated based on the rule's confidence, e.g., $2 \cdot conf_{rule} - 1$ in our framework.

The measure of ModelScore in Eq. (5) prefers abduced labels not far from the model's current prediction. The measure of KBScore in Eq. (6) shares a similar form of weighted maximum satisfiability (Weighted MAX-SAT) problem, where it prefers abduced labels that have a small combined weight of inconsistent rules. The weighting coefficient α in Eq. (4) combines these two scores to form the final consistency measure. When optimizing Eq. (4), if the search space is large, we can solve with derivative free optimization [Yu *et al.*, 2016]. Otherwise, as in this work, we can directly search all candidates.

Our proposed consistency measure is general, which can be regarded as a generalization of previous measures:

Proposition 1. *The consistency measure of previous abductive learning approaches [Dai* et al.*, 2019; Cai* et al.*, 2021; Huang* et al.*, 2021b] is a special case of Eq. (4) where* $\mathrm{KBScore} = 0$.

This conclusion is clear because previous methods require the abduced labels to be consistent with the whole knowledge base, and therefore there are no inconsistent ground rules.

Method	Accuracy	ABL-KG	Accuracy
RF	0.859±0.077	w/o Rem	0.890±0.046
PL	0.862±0.075	w/o Contr	0.773±0.063
Tri	0.865±0.093	w/o Post	0.878±0.066
ABL-KG	**0.915±0.042**	ABL-Expert	0.929±0.035

Table 1: Test accuracy of different methods (left) and ablation studies (right) on animal classification task. "Rem", "Contr" and "Post" denote "remembering", "contradiction handling" and "post-processing", respectively. "Expert" means domain knowledge from an expert.

Step	Origin	Mined	Rem (w/ Contr)	Post
# Triplets/Rules	34M	44214	51	40
Abduced Acc.	-	85.9%	90.5%	94.2%

Table 2: Statistics of rules after each step in ABL-KG. "Origin" contains triplets and the others contain rules.

5 Experiments

This section presents the experimental results on three different types of tasks, including a task on tabular data, two tasks on graph data and a task on image data, to demonstrate that ABL-KG is a general framework that can automatically extract domain knowledge rules from large-scale KGs and leverage them for improving the performance of machine learning models. The code is available for download[1].

5.1 Animal Classification

The zoo animal classification dataset [Dua and Graff, 2017] contains animals' attributes (e.g., backbone, legs) and their categories (e.g., bird, fish), along with the names of each attribute and class (task specification). In our experiment, we use only one labeled sample of each class for model initialization. The remaining is randomly split into 70% unlabeled data and 30% test data. We use ConceptNet [Speer *et al.*, 2017], a large-scale semantic network with 34 million edges, as our KG. We use GloVe [Pennington *et al.*, 2014] for signature matching (cf. Section 4.4). Rules are mined using our algorithm in Section 4.3, where d is set to 2, and we set the confidence of IsA relation to be 1.0 and others 0.9.

The left part of Table 1 shows the test accuracy. Here, the random forest (RF) [Breiman, 2001] is used as the classifier in all methods. ABL-KG is compared with two types of baselines, i.e., RF using only labeled data and semi-supervised learning methods leveraging unlabeled data, including Pseudo-Label (PL) [Lee, 2013] and Tri-training (Tri) [Zhou and Li, 2005]. It is obvious that the performance of ABL-KG is significantly superior to other approaches. Intriguingly, the performance of ABL-KG is close to that of ABL with manually designed rules, which take an expert about one hour to write. Since all rules come from the mining of ConceptNet, it demonstrates that ABL-KG is able to leverage a large-scale and noisy KG for the training of a machine learning model.

The ablation studies are presented on the right part of Table 1. The performance of ABL-KG drops if skipping any of the "remembering", "contradiction handling" and "post-processing" steps, indicating that all procedures play a significant role in ABL-KG. Specifically, without "contradiction handling", ABL-KG reaches the lowest performance. We check the generated rules in this case, and find that they contain some false rules, e.g., "$\neg bird(X)$" ("nothing is a bird"), which would result in great performance degradation. Further analysis reveals that this is caused by forgetting $animal$ on two contradictions from KG, i.e., "$animal(X) \leftarrow bird(X)$" and "$\neg animal(X) \leftarrow bird(X)$". Contradiction handling would remove these contradictory rules and avoid this error.

We analyze the rules after each step, as shown in Table 2. Starting from 34 million triplets, ABL-KG gradually mines useful rules and drops irrelevant ones, leading to the improvement of abduced label accuracy. We compare the mined rules with the expert's ones. Although the former lacks some domain knowledge due to KG incompleteness, most of the rules written by the expert have been discovered by ABL-KG.

[1]https://github.com/AbductiveLearning/ABL-KG

Method	Hits@1	Hits@10	MRR
AlignE	0.504±0.013	0.863±0.007	0.626±0.010
AlignE+	0.580±0.014	0.890±0.005	0.676±0.011
AlignE++	0.615±0.011	0.895±0.005	0.706±0.019
ABL-KG	**0.645±0.005**	**0.891±0.004**	**0.732±0.004**
ABL-Expert	0.688±0.005	0.905±0.002	0.765±0.004

Table 3: Entity alignment results on DBP-EN-FR.

5.2 Entity Alignment

The task seeks to find the identical entities from two KGs (input data). The entity alignment model learns embeddings for the two KGs to measure entity similarity. In our experiment, we consider the widely-used cross-lingual dataset DBP-EN-FR, which was proposed in the OpenEA benchmark study [Sun *et al.*, 2020]. It aims to align the entities in the English and French DBpedia [Lehmann *et al.*, 2015]. It has 15000 entity alignment pairs and we use 20% of them as training data. We choose the popular entity alignment model AlignE [Sun *et al.*, 2018] as the basic aligner in our experiment, and implement two semi-supervised variants, i.e., AlignE+ and AlignE++, as baselines. AlignE+ employs self-training [Yarowsky, 1995] and selects the predicted entity alignment pairs whose embedding similarity is greater than 0.9 to augment training data. AlignE++ improves AlignE+ by using entity dependencies to reduce the noise in augmented training data [Liu *et al.*, 2023]. We use Wikidata [Vrandecic and Krötzsch, 2014] and YAGO3 [Rebele *et al.*, 2016] as the background KGs to acquire a $\mathcal{KB}$ for entity alignment, and use GloVe [Pennington *et al.*, 2014] for relation matching during remembering.

We report the Hits@1, Hits@10, and MRR results of five-fold cross-validation in Table 3. Our ABL-based methods, ABL-KG and ABL-Expert, bring significant improvement to AlignE. They also outperform the semi-supervised variants. This is because by leveraging mined/manual knowledge base, abductive reasoning can identify and resolve the alignment inconsistency issue, which improves the quality of new training data. An example of the automatically mined rules in ABL-KG is "$sameAs(X,Y) \leftarrow spouse(X,A) \wedge father(C,B) \wedge$

Method	Hits@1	Hits@10	MRR
TransE	0.199±0.002	0.500±0.004	0.300±0.003
TransE+	0.183±0.002	0.508±0.004	0.294±0.002
ABL-KG	**0.216±0.006**	**0.551±0.004**	**0.334±0.004**
ABL-Expert	0.235±0.004	0.557±0.002	0.348±0.003

Table 4: Link prediction results on FB15K-237.

$mother(C,Y) \wedge sameAs(A,B)$", which means that a person's father and mother are spouses. In the abductive learning stage, these rules form a logic program that can check the dependency of predicted entity alignment pairs and resolve alignment inconsistency. ABL-Expert achieves better performance than ABL-KG, because the expert rules are superior to our automatic rule mining method in terms of rule noise.

5.3 Link Prediction

Link prediction is the task of predicting the missing entity of an incomplete triple like (IJCAI-2023, $\texttt{location}$, ?). We consider the benchmark dataset FB15K-237 [Toutanova and Chen, 2015] and choose the most popular model, TransE [Bordes *et al.*, 2013], as the basic learning model in the experiment. Given a triplet, TransE interprets the relation as a translation vector between the subject and object. The translation error is defined as the energy of the triplet. We also use self-training to implement a semi-supervised baseline, denoted by TransE+. We mine rules from DBpedia [Lehmann *et al.*, 2015] and use GloVe [Pennington *et al.*, 2014] for relation matching in the remembering process to build a $\mathcal{KB}$ targeted to FB15K-237.

We follow the training/validation/test data splits of FB15K-237, and report the average test results of five runs in Table 4. We find that the semi-supervised method TransE+ even underperforms TransE. This is because link prediction is a difficult task and the low accuracy results in much noise in the abduced data, which hurts the model training. By contrast, both ABL-KG and ABL-Expert offer performance improvement. The rules extracted from other background KGs can provide additional capability for accurately inferring new triplets and correcting incorrectly predicted triplets. It is noteworthy that the performance of ABL-KG is close to that of ABL-Expert, showing that the automatically mined rules in ABL-KG have a similar quality to the expert's rules, which could reduce the human efforts on knowledge engineering.

5.4 Image Classification

The task's input is images from ADE20K [Zhou *et al.*, 2017]. We reform it as a multi-label classification task, which requires a model to determine the objects (e.g., bed, sidewalk) in images. From totally 150 categories of objects, we randomly select 12 frequently appearing objects as labels. In our task, only 5% or 10% of all 20k images are labeled. Each image, labeled or not, comes with a scene description (e.g., bedroom, street), with 1055 possible scenes. ABL-KG uses ConceptNet [Speer *et al.*, 2017] as the KG, and the confidence of mined rules is set to 1.0 for the IsA relation and 0.8 for the rest. In addition, we use an initial knowledge base that contains only 40 rules to cover the incompleteness of ConceptNet, and the final $\mathcal{KB}$ contains 590 rules. All methods

Method	5%	10%
ResNet	0.604±0.022	0.646±0.016
PL	0.612±0.028	0.653±0.014
VAT	0.620±0.015	0.669±0.008
ABL-Init	0.613±0.029	0.654±0.020
ABL-KG	**0.649±0.018**	**0.680±0.014**

Table 5: F1-score of image classification with different label rates.

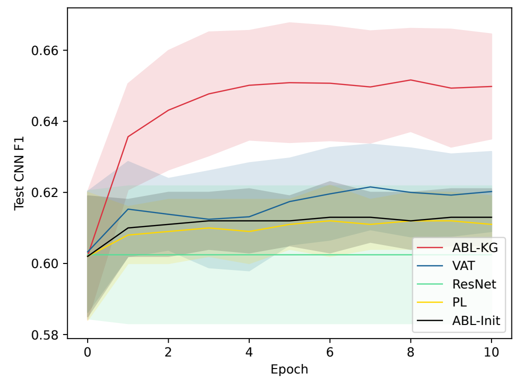

Figure 2: Learning curves on the image classification task. Shaded regions represent standard deviation.

use a ResNet-50 [He *et al.*, 2016] pre-trained on ImageNet as the learning model, in which the last layer is replaced to fit the multi-label task, and the model is fine-tuned on the given labeled and abduced labels.

Table 5 presents the F1-scores (macro-averaging) of compared approaches, including supervised (ResNet [He *et al.*, 2016]) and semi-supervised methods (Pseudo-Label (PL) [Lee, 2013] and Virtual Adversarial Training (VAT) [Miyato *et al.*, 2018]). With different label rates, ABL-KG achieves the highest F1. When there are fewer labeled data, the performance gain of ABL-KG compared with supervised methods is greater. An ablation study only using the initial knowledge base (ABL-Init) indicates that the performance gain mainly comes from the mined KG rules. The learning curves of various approaches are shown in Figure 2. Starting with the same performance, ABL-KG achieves higher accuracy than other methods.

6 Conclusion

Previous works that integrate machine learning and logical reasoning usually require a manually engineered knowledge base. To reduce this burden, we propose to exploit existing large-scale knowledge graphs as knowledge resources. In this paper, we propose the ABL-KG framework which can extract an interpretable knowledge base automatically in the form of logic rules and use them for abductive learning.

Experiments on various tasks validate that the extracted knowledge contributes to an improvement of machine learning models under the abduction strategy that employs the consistency measure designed for noisy and uncertain rules. The limitation of ABL-KG is that it assumes the existing knowledge graphs contain the needed knowledge, which may not always be sufficient due to the limited expressive power of triplets. In future work, it would be intriguing to exploit other knowledge representation forms such as ontology.

Acknowledgments

This research was supported by NSFC (61921006, 62272219, 62206124).

References

[Bengio, 2017] Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795. 2013.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Cai *et al.*, 2021] Le-Wen Cai, Wang-Zhou Dai, Yu-Xuan Huang, Yu-Feng Li, Stephen Muggleton, and Yuan Jiang. Abductive learning with ground knowledge base. In *IJCAI*, pages 1815–1821, 2021.

[Carlson *et al.*, 2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, pages 1306–1313, 2010.

[Dai and Muggleton, 2021] Wang-Zhou Dai and Stephen H. Muggleton. Abductive knowledge induction from raw data. In *IJCAI*, pages 1845–1851, 2021.

[Dai *et al.*, 2019] Wang-Zhou Dai, Qiuling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. In *NeurIPS*, pages 2811–2822. 2019.

[d'Amato *et al.*, 2016] Claudia d'Amato, Steffen Staab, Andrea GB Tettamanzi, Tran Duc Minh, and Fabien Gandon. Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 333–338, 2016.

[De Raedt and Kimmig, 2015] Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100(1):5–47, 2015.

[Delgrande, 2014] James P Delgrande. Towards a knowledge level analysis of forgetting. In *KR*, pages 606–609, 2014.

[Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2017.

[Eiter and Kern-Isberner, 2019] Thomas Eiter and Gabriele Kern-Isberner. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI-Künstliche Intelligenz*, 33(1):9–33, 2019.

[Galárraga *et al.*, 2013] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422, 2013.

[Galárraga *et al.*, 2015] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *The VLDB Journal*, 24(6):707–730, 2015.

[Garcez *et al.*, 2019] A Garcez, M Gori, LC Lamb, L Serafini, M Spranger, and SN Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *Journal of Applied Logics*, 6(4):611–632, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hogan *et al.*, 2022] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys*, 54(4):71:1–71:37, 2022.

[Huang *et al.*, 2020] Yu-Xuan Huang, Wang-Zhou Dai, Jian Yang, Le-Wen Cai, Shaofen Cheng, Ruizhang Huang, Yu-Feng Li, and Zhi-Hua Zhou. Semi-supervised abductive learning and its application to theft judicial sentencing. In *ICDM*, pages 1070–1075, 2020.

[Huang *et al.*, 2021a] Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. In *NeurIPS*, pages 25134–25145. 2021.

[Huang *et al.*, 2021b] Yu-Xuan Huang, Wang-Zhou Dai, Le-Wen Cai, Stephen H Muggleton, and Yuan Jiang. Fast abductive learning by similarity-based consistency optimization. In *NeurIPS*, pages 26574–26584. 2021.

[Huang *et al.*, 2023] Yu-Xuan Huang, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. Enabling knowledge refinement upon new concepts in abductive learning. In *AAAI*, page to appear, 2023.

[Ji *et al.*, 2022] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[Koller *et al.*, 2007] Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, David Heckerman, Chris Meek, et al. *Introduction to statistical relational learning*. The MIT Press, 2007.

[Lee, 2013] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[Lehmann *et al.*, 2015] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[Li *et al.*, 2020] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *ICML*, pages 5884–5894, 2020.

[Lifschitz, 2002] Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1-2):39–54, 2002.

[Lin and Reiter, 1994] Fangzhen Lin and Ray Reiter. Forget it! In *Proceedings of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.

[Liu *et al.*, 2023] Bing Liu, Tiancheng Lan, Wen Hua, and Guido Zuccon. Dependency-aware self-training for entity alignment. In *WSDM*, pages 796–804, 2023.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, pages 3749–3759. 2018.

[Marino *et al.*, 2017] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, pages 2673–2681, 2017.

[Miyato *et al.*, 2018] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

[Nickel *et al.*, 2015] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Raedt *et al.*, 2016] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

[Raedt *et al.*, 2020] Luc De Raedt, Sebastijan Dumancic, Robin Manhaeve, and Giuseppe Marra. From statistical relational to neuro-symbolic artificial intelligence. In *IJCAI*, pages 4943–4950, 2020.

[Rebele *et al.*, 2016] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web*, pages 177–185, 2016.

[Schreiber and Raimond, 2014] Guus Schreiber and Yves Raimond. *RDF 1.1 Primer*. W3C Working Group Note. World-Wide Web Consortium, 2014.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017.

[Sun *et al.*, 2018] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402, 2018.

[Sun *et al.*, 2020] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326–2340, 2020.

[Toutanova and Chen, 2015] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.

[Unger *et al.*, 2012] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over RDF data. In *WWW*, pages 639–648, 2012.

[Vrandecic and Krötzsch, 2014] Denny Vrandecic and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[Wang *et al.*, 2005] Kewen Wang, Abdul Sattar, and Kaile Su. A theory of forgetting in logic programming. In *AAAI*, pages 682–688, 2005.

[Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[Wang *et al.*, 2018] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, pages 6857–6866, 2018.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, pages 189–196, 1995.

[Yu *et al.*, 2016] Yang Yu, Hong Qian, and Yi-Qi Hu. Derivative-free optimization via classification. In *AAAI*, pages 2286–2292, 2016.

[Zhou and Huang, 2022] Zhi-Hua Zhou and Yu-Xuan Huang. Abductive learning. In Pascal Hitzler and Md. Kamruzzaman Sarker, editors, *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 353–369. IOS Press, Amsterdam, 2022.

[Zhou and Li, 2005] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[Zhou *et al.*, 2017] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 5122–5130, 2017.

[Zhou, 2019] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62(7):76101, 2019.