Enabling Knowledge Refinement upon New Concepts in Abductive Learning

Yu-Xuan Huang, Wang-Zhou Dai, Yuan Jiang, Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{huangyx, daiwz, jiangy, zhouzh}@lamda.nju.edu.cn

Abstract

Recently there are great efforts on leveraging machine learning and logical reasoning. Many approaches start from a given knowledge base, and then try to utilize the knowledge to help machine learning. In real practice, however, the given knowledge base can often be incomplete or even noisy, and thus, it is crucial to develop the ability of knowledge refinement or enhancement. This paper proposes to enable the Abductive learning (ABL) paradigm to have the ability of knowledge refinement/enhancement. In particular, we focus on the problem that, in contrast to closed-environment tasks where a fixed set of symbols are enough to represent the concepts in the domain, in open-environment tasks new concepts may emerge. Ignoring those new concepts can lead to significant performance decay, whereas it is challenging to identify new concepts and add them to the existing knowledge base with potential conflicts resolved. We propose the ABL$_{nc}$ approach which exploits machine learning in ABL to identify new concepts from data, exploits knowledge graph to match them with entities, and refines existing knowledge base to resolve conflicts. The refined/enhanced knowledge base can then be used in the next loop of ABL and help improve the performance of machine learning. Experiments on three neuro-symbolic learning tasks verified the effectiveness of the proposed approach.

1 Introduction

Integrating data-driven machine learning and knowledge-driven reasoning in a unified framework is considered to be one of the keys to the next generation of Artificial Intelligence (AI). Recent years have witnessed representative progress in this direction such as Neuro-Symbolic (NeSy) Learning (Garcez et al. 2019; Raedt et al. 2020) and Statistical Relational AI (StarAI) (Raedt et al. 2016). Most of them try to build a neural network structure or a probabilistic graphical model based on domain knowledge expressed in first-order logic. Probabilistic Logic Program (PLP) (De Raedt and Kimmig 2015) extends first-order logic to accommodate a probability distribution in Herbrand Universe to conduct probabilistic inference.

Recently, some researchers propose to build a hybrid model, which usually consists of a perception model for machine learning and a reasoning model for logical reasoning. Representative frameworks include DeepProbLog (Manhaeve et al. 2018) and ABductive Learning (ABL) (Zhou 2019; Zhou and Huang 2022), in which the perception model learns to convert raw data into primitive logic facts, serving as input to symbolic reasoning; while the reasoning model tries to infer the truth-value of the perceived logic facts based on a given knowledge base, for updating the perception model. The integration of the two systems is realized by abduction, also known as abductive reasoning.

Most of the above methods focus on closed-environment tasks, where a fixed and complete set of symbols representing the primitive concepts in domain are predefined in the knowledge base. In other words, they assume that the knowledge base has contained all possible primitive concepts in the task, and the logic facts to be perceived from incoming data must be an instance of these predefined concepts. For example, when trying to learn whether a handwritten arithmetic equation is correct, the system is already provided with all the symbols of the numbers and operator symbols that could appear in the data.

In real-world applications, the environment we encounter is usually open and dynamic, a major challenge of which is the emerging new concepts (Zhou 2022). Hence, it is crucial for machine learning to enquire the ability to identify and internalize the emerging new concepts, otherwise the model would treat all the instances of new concepts as known ones and degrades the performance.

The first difficulty lies in how to detect new concepts while preserving the model performance on known concepts (Zhou 2022). The old background knowledge needs to be enhanced and refined to incorporate the discovered new concepts, whose semantics will be expressed by a set of newly induced/abduced logic clauses that may conflict with the current background knowledge base. Furthermore, researchers have pointed out that adding arbitrary new rules to an existing knowledge base might harm the interpretability of the model (Ai et al. 2021). Therefore, the semantics of the new concepts should be reflected in their names with meaningful words instead of meaningless symbols for better human understanding.

This paper proposes to tackle the above challenges with ABL$_{nc}$ (ABductive Learning with New Concept). It first leverages machine learning to monitor the distribution change in raw feature space and identify the instances of new concepts. Then it conducts a rule induction process to figure out the relations between the detected new concepts and the known concepts. A conflict resolution process is taken when there is a conflict between the induced rules and knowledge base, after which the rules involving the new concepts are merged into the knowledge base. Then ABL_{nc} continues abductive learning to improve the performance of machine learning by exploiting the refined knowledge base. The above routine repeats iteratively. As a post-learning process, ABL_{nc} tries to match the new concepts to corresponding entities in a large-scale commonsense knowledge graph.

We verify the effectiveness of ABL_{nc} on three neuro-symbolic tasks. Experimental results show that ABL_{nc} can learn correct definitions of the new concepts from data and resolve the conflicts in knowledge base. The performance of perception model also improves during abductive learning with the refined knowledge base. Moreover, the new concepts could be matched to proper entities in knowledge graphs, which assign human-understandable names instead of denoting them with meaningless symbols.

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 representative paradigms for unifying machine learning and logical reasoning. PLP tries to extend first-order logic to accommodate probabilistic groundings such that probabilistic inference can be conducted; SRL attempts to use domain knowledge expressed in first-order logic to construct a probabilistic graphical model structure for statistical inference. Neuro-Symbolic (NeSy) Learning (Garcez et al. 2019; Raedt et al. 2020) shares the same motivation with SRL, where the external domain knowledge is used for building an explainable neural structure. Recently, some novel approaches that try to build a hybrid model have been proposed, including DeepProbLog (Manhaeve et al. 2018), Abductive Learning (ABL) (Dai et al. 2019; Zhou 2019) and Neural-Grammar-Symbolic model (Li et al. 2020).

Recent progress on ABL (Dai and Muggleton 2021) has shown its capability of rule induction with incomplete background knowledge, in which the set of primitive logic facts is fixed and manually predefined. In contrast, ABL_{nc} tries to solve a totally different type of problem in which: 1) the new concepts to be discovered are undefined in the knowledge base; 2) the new concepts could appear in the perception level and have to be discovered by the perception model; 3) the incorporation of the newly induced rules into the old knowledge base involves conflict resolution.

Inductive Logic Programming (ILP) (Muggleton and De Raedt 1994) is a subfield of symbolic artificial intelligence, where the goal is to learn a logic theory that generalizes given training examples. ILP can learn human-readable hypotheses from small amounts of data in the form of logic program. Representative ILP systems include ILASP (Law, Russo, and Broda 2015), Metagol (Muggleton, Lin, and Tamaddoni-Nezhad 2015), ALEPH (Srinivasan 2001) and the recently proposed Popper (Cropper and Morel 2021). These ILP systems are designed for learning from symbolized data. Otherwise, they need to use fully trained machine learning models to extract symbols from the raw inputs.

The exploitation of incremental learning has been studied in both machine learning and ILP. In machine learning which usually involves sub-symbolic data, incremental learning aims to update the models from data stream sequentially and has achieved many successes (Zhou and Chen 2002; Masana et al. 2020). Learning with emerging new classes is a kind of incremental learning, where a lot of approaches have been developed (Ma and Perkins 2003; Mu, Ting, and Zhou 2017; Zhang et al. 2020). In ILP which processes symbolic inputs, this process is also known as Theory Revision (Wrobel 1996; Esposito et al. 2000) or Bias Reformulation (Lin et al. 2014), where the learner alters previously inferred knowledge to fit new observations in order to exploit previous computations. Our approach handles raw input in sub-symbolic representations, and requires learning symbolic logic programs for new concepts, which cannot be directly solved by previous methods.

Knowledge graphs (KG) are structured representations of human knowledge that model information in the form of a graph. We use the RDF (Resource Description Framework) KGs (Schreiber and Raimond 2014), which contain triplets of entities (nodes) and relationships between them (edges). Recently, representation learning has shown its ability to capture KG semantics in vector space, and uses the acquired embeddings for downstream tasks such as link prediction (Wang et al. 2017). Note that although KGs may contain desired knowledge about unknown concepts, we could not directly exploit them because most of them are irrelevant and will result in low reasoning efficiency.

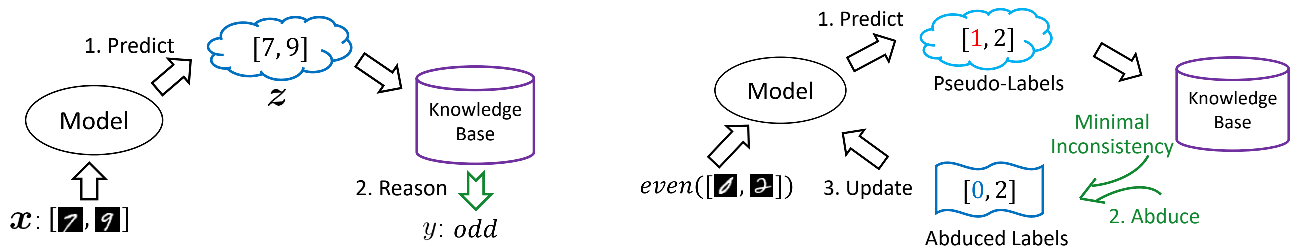
3 Abductive Learning

3.1 Inference

The ABductive Learning (ABL) (Zhou 2019; Zhou and Huang 2022) framework consists of a perception model f and a reasoning model. The perception model f maps the raw input data $\boldsymbol{x}$ into discrete symbols $\boldsymbol{z}$. The reasoning model contains a knowledge base KB consisting of first-order logic rules, which receives $\boldsymbol{z}$ and inferences the final output y by logical reasoning. For example, given the rules of even and odd numbers, the reasoning model would output a ground fact $y = odd$ when the input fact is $\boldsymbol{z} = [7, 9]$, and $y = even$ when $\boldsymbol{z} = [0, 2]$. Figure 1(a) gives an example of the inference process in the experiment (cf. Section 5.3).

3.2 Learning

Formally, given unlabeled data $\boldsymbol{X} = \{\boldsymbol{x}^{\langle 1\rangle}, \boldsymbol{x}^{\langle 2\rangle}, \ldots\}$, knowledge base KB and the final desired output $Y = \{y^{\langle 1\rangle}, y^{\langle 2\rangle}, \ldots\}$, we are required to learn f that predicts the labels $\boldsymbol{z}$ of the $\boldsymbol{x}$ that together with KB entail y. ABL first obtains the symbolic predictions $\boldsymbol{z} = f(\boldsymbol{x})$ as pseudo-labels. Then the reasoning model with KB tries to revise the pseudo-labels $\boldsymbol{z}$ by abduction (i.e., abductive reasoning), a basic form of logical inference that seeks an explanation for observed phenomena. Finally, ABL uses the abduced labels $\bar{\boldsymbol{z}}$ to update f, and the above routine repeats iteratively.



(a) Inference.

(b) Learning.

Figure 1: An example of the inference and learning process of ABL. In the inference stage, the input data is a list of digit images, and the final output is the reasoning result of predicted symbols. In the learning stage, the reasoning model abduces revised pseudo-labels, which are then used to update the perception model.

For example, consider the knowledge base in Figure 1(b) containing the following logic rule:

$$even([X_1, X_2]) \leftarrow divisible(X_1, 2) \wedge divisible(X_2, 2), \tag{1}$$

where "$\wedge$" denotes conjunction (and); "$\leftarrow$" is implication, which means that if condition (body) on the right of "$\leftarrow$" holds, then consequent (head) on the left holds. Rule (1) means $[X_1, X_2]$ is a list of (single-digit) even numbers if both X_1 and X_2 are divisible by 2. The pseudo-labels $\boldsymbol{z} = [1, 2]$ in Figure 1(b) are inconsistent with KB, since they do not entail $even$. Then the reasoning model revises $\boldsymbol{z}$ to $\bar{\boldsymbol{z}} = [0, 2]$ by abduction, which are used to update f.

4 The ABL$_{nc}$ Approach

In this section, we introduce our ABL$_{nc}$ (ABductive Learning with New Concept) method, which tackles the emerging new concepts in data under the ABL framework.

4.1 Problem Setting

We are given a knowledge base KB containing domain knowledge about known concepts, an initial perception model f that has been trained with some labeled data of the known concepts, and a large-scale knowledge graph KG. Each pseudo-label concept $\boldsymbol{z} \subseteq \mathcal{Z}$ of the perception model f can be represented by a constant (e.g., 9) or a ground atom (e.g., $nine(image)$) in KB, while each target concept $y \in \mathcal{Y}$ of the reasoning model is represented by a predicate defined by first-order logic rules in KB, e.g., $even(X)$. Each example $\boldsymbol{x} \in \boldsymbol{X}$ comes with the corresponding target output $y \in \boldsymbol{Y}$, with *no* supervision on pseudo-labels $\boldsymbol{z}$.

The new concepts come in the *learning* stage, after which the model f and knowledge base KB could be refined to adapt to the new environment. Since new concepts might emerge while performing perception or reasoning, we split the problem into two cases: 1) new concepts emerge at perception level and 2) new concepts emerge at reasoning level. Figure 2 shows the setting of ABL$_{nc}$.

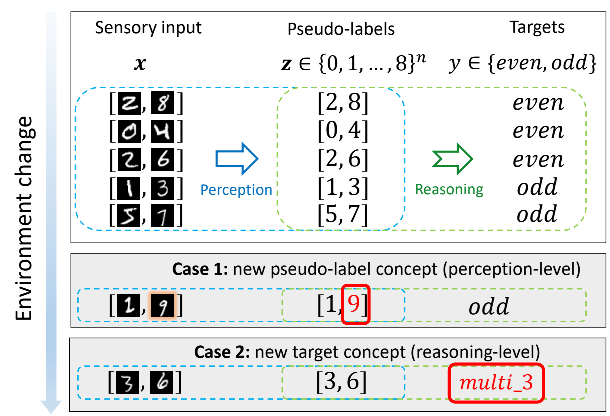


Figure 2: The setting of ABL$_{nc}$, where new concepts could emerge at perception or reasoning level.

New Concepts at Perception Level. When a new concept emerges at perception level, it leads to an augmented pseudo-label space $\mathcal{Z} = \mathcal{Z}_{new} \cup \mathcal{Z}$ of the perception model f, which classifies instances either to one of the known or the new concepts. In this setting, the ABL$_{nc}$ is required to:

- Update f to accurately predict known and new concepts
- Learn rules about the new concepts and refine KB
- Match new concepts to corresponding entities in KG

For example, encountering a new concept 9 in Figure 2, we have to update perception model f to accurately recognize it, and refine logic rules in knowledge base KB so that it could conduct logical reasoning correctly. The concepts of new digits would serve as attributes in the body of corresponding logic rules. The third goal is necessary for users to better understand the new concepts, e.g., $number\_9$ conveys more information than a meaningless symbol (e.g., new). Furthermore, the matched entity could bring external knowledge from KG to improve KB.

New Concepts at Reasoning Level. Similarly, an emerging new concept at reasoning level requires an augmented label space $\mathcal{Y} = \mathcal{Y}_{new} \cup \mathcal{Y}$ of the reasoning model, which conducts logical reasoning with KB and entails ground facts of a known concept or the new concept. The goals are similar to the previous setting, where the first goal becomes updating f using unlabeled data $\boldsymbol{X}$. The new concepts would be target predicates in the head of logic rules. Note that this setting is different from the rule induction problem in (Dai and Muggleton 2021) since the new concepts $\mathcal{Y}_{new}$ may conflict with existing $\mathcal{Y}$ and thus a theory revision to KB is required.

These two tasks turn out to be quite challenging: on the one hand, it is difficult for the perception model f to perform correct classification with no labeled data of new concepts; on the other hand, the knowledge base KB contains zero knowledge about new concepts and hard to conduct logic inference. Moreover, it usually happens that the learned rules of new concepts may *conflict* with the original KB.

4.2 ABL$_{nc}$ Overview

We propose the ABL$_{nc}$ approach, which handles new concepts at perception level and reasoning level in a unified way. The learning involves three components: new concepts detection, knowledge refinement and abductive learning, followed by a post-learning process of KG concept matching.

New Concepts Detection. For the new concepts emerging at perception level, the model f would inevitably make wrong predictions due to the absence of labeled data. In this situation, ABL$_{nc}$ first exploits a new class detection model g to roughly identify new concepts from data, which serve as labeled data of new concepts to update f. In this step, the pseudo-labels $\boldsymbol{z}$ of instance $\boldsymbol{x}$ are calculated as:

$$\boldsymbol{z} = \begin{cases} new, & g(\boldsymbol{x}) = 1 \\ f(\boldsymbol{x}), & g(\boldsymbol{x}) = 0 \end{cases} \tag{2}$$

where new is the label of new concepts, and $g(\boldsymbol{x})$ indicates whether $\boldsymbol{x}$ is identified as a new concept. Note that since the new class detector g might misclassify known concepts as new ones, it is only used in the first ABL iteration.

Knowledge Refinement. The knowledge refinement involves *rule learning* and *conflict resolution*. Getting the pseudo-labels $\boldsymbol{Z}$ and final output Y, ABL$_{nc}$ first tries to learn rules R about the new concepts. Since R and KB are both logic programs, the task could be formulated as Inductive Logic Programming (ILP) (Muggleton and De Raedt 1994). If new concepts emerge at perception level, ABL$_{nc}$ treats samples with the same final output y as positive examples and the rest as negative ones. For new concepts at reasoning level, ABL$_{nc}$ regards samples with new concepts $y \in \mathcal{Y}_{new}$ as positive examples and the rest as negative.

The learned rules R, especially the ones describing new concepts at reasoning level, may conflict with KB's rules of known concepts. ABL$_{nc}$ employs conflict detection between R and KB. If a conflict happens, a *conflict resolution* procedure is conducted, after which KB is refined by rules of new concepts. Note that inserting the R into KB may damage performance if R is low-quality. In this situation, ABL$_{nc}$ could resort to human experts for manual checking.

Abductive Learning. The last step involves abductive learning (Zhou 2019; Zhou and Huang 2022). The reasoning model in ABL$_{nc}$ tries to revise the pseudo-labels $\boldsymbol{Z}$ by abduction based on the refined KB_{new}. The abduction is conducted based on the principle of minimal inconsistency between revised pseudo-labels $\bar{\boldsymbol{Z}}$ and KB_{new}, where various consistency measure (Dai et al. 2019; Huang et al. 2020; Cai et al. 2021; Huang et al. 2021) could be used. Then the revised labels are used as ground-truth to update f.

The above three steps repeat iteratively. Algorithm 1 shows an outline of ABL$_{nc}$. By revising the pseudo-labels by learned rules R, the quality of pseudo-labels is improved, leading to a better performance of the perception model f. The perception model f in turn produces higher quality pseudo-labels, which benefit the learning of R. Therefore, by model update and knowledge refinement, the perception and reasoning model in ABL$_{nc}$ could benefit each other and tackle the arrival of new concepts. The last step of ABL$_{nc}$ involves matching new concepts in knowledge graph.

Algorithm 1 ABL$_{nc}$

Input: Perception model f; New class detection model g; Knowledge base KB; Knowledge graph KG; Unlabeled data $\boldsymbol{X}$; Desired output Y

Output: Perception model f; Knowledge base KB_{new}; new concepts name c

for $i = 1$ **to** $turn\ limit$ **do**
  if $i == 1$ and new concepts at perception level **then**
    $\boldsymbol{Z}$ = merge $g(\boldsymbol{X})$ and $f(\boldsymbol{X})$ based on Eq. (2)
  else
    $\boldsymbol{Z} = f(\boldsymbol{X})$
  end if
  $R = \mathrm{LearnRule}(\boldsymbol{Z}, Y)$ # Rules of new concepts
  if $\mathrm{ConflictDetection}(KB, R)$ **then**
    $KB_{new} = \mathrm{ConflictResolution}(KB, R)$
  else
    $KB_{new} = KB \cup R$
  end if
  $\bar{\boldsymbol{Z}} = \mathrm{Abduce}(KB_{new}, \boldsymbol{Z}, Y)$
  $f = \mathrm{Update}(f, \boldsymbol{X}, \bar{\boldsymbol{Z}})$
end for
$c = \mathrm{MatchKG}(KG, \boldsymbol{Z}, R)$
return f, KB_{new}, c

4.3 Conflict Resolution

Conflict Detection. Without loss of generality, we assume that the head of each rule represents the output concept $y \in \mathcal{Y}$, e.g., $even(X)$.

Definition 1 (Conflict Rules) *Two rules r_1 and r_2 are conflict rules, if (i) their head predicates represent different concepts y, and (ii) there exists a sample $\boldsymbol{z}$ such that the head of r_1 and r_2 hold at the same time.*

For example, consider the rule of new concepts "multiples of three" (all digits in the list are multiples of 3):

$$new([X_1, X_2]) \leftarrow divisible(X_1, 3) \wedge divisible(X_2, 3). \tag{3}$$

It conflicts with a known rule of $even$ (all digits are even):

$$even([X_1, X_2]) \leftarrow divisible(X_1, 2) \wedge divisible(X_2, 2), \tag{4}$$

Algorithm 2 Conflict Resolution

Input: Knowledge base KB; Rule set R
Output: Knowledge base KB_{new}

for $r_{new} \in R$ with new concepts in rule **do**
  for $r_{known} \in KB$ with known concepts in rule **do**
    if $\mathrm{head}(r_{new}) == \mathrm{head}(r_{known})$ **then**
      continue
    end if
    $conflict = (\mathrm{body}(r_{new}) \wedge \mathrm{body}(r_{known}))$ # An example has multiple facts in $\mathcal{Y}$
    if $\mathrm{Satisfiable}(KB \cup conflict)$ **then**
      $r_{known} = (head(r_{known}) \leftarrow body(r_{known}) \wedge \neg body(r_{new}))$
    end if
  end for
end for
$KB_{new} = KB \cup R$
return KB_{new}

because if $[X_1, X_2] = [6, 6]$, the new concept $y = new$ and the known concept $y = even$ hold simultaneously, causing a conflict because there are only one ground fact y of each example. In this case, the desired output should be $y = new$ because the new concepts have higher priority.

For conflict detection (cf. Line 8 in Algorithm 1), we check if there exists an input $\boldsymbol{z}$ that satisfies two different target concepts $y_1, y_2 \in \mathcal{Y}$ simultaneously. Note that if we allow an example to be satisfied by multiple target concepts (i.e., adopting the *multi-label classification* setting), the conflict detection and resolution processes could be skipped.

Conflict Resolution Algorithm. We propose a conflict resolution algorithm shown in Algorithm 2. Since KB contains correct domain knowledge of known concepts, the conflict only exists between R and KB. If a new rule r_{new} and a known rule r_{known} cause a conflict, we resolve it by adding the negation of r_{new}'s body to the body of r_{known}, i.e., a kind of specialization operator (Muggleton and De Raedt 1994). Note that in this step, a substitution operation is required to replace the variables of the added body. Following the above example, after simplification, the rule of $even$ would be changed to:

$$even([X_1, X_2]) \leftarrow divisible(X_1, 2) \wedge divisible(X_2, 2) \wedge \neg divisible(X_1, 3). \tag{5}$$

$$even([X_1, X_2]) \leftarrow divisible(X_1, 2) \wedge divisible(X_2, 2) \wedge \neg divisible(X_2, 3). \tag{6}$$

Proposition 1 *No conflict rules exist in KB_{new} after Algorithm 2.*

The conclusion is clear because the algorithm modifies r_{known}'s body such that the bodies of r_{new} and r_{known} cannot cover the same examples.

4.4 New Concept Matching in KG

Algorithm 3 shows the procedure of matching new concepts in knowledge graph (KG). To find an entity in a large-scale knowledge graph KG that matches the properties of the new concepts, it is unnecessary and inefficient to use the entire KG. ABL$_{nc}$ first extracts a sub-graph KG' of KG by Breadth-First Search (BFS) with depth d, starting from an entity node that represents the task, e.g., $number$ or $chess$ in our experiments.

Algorithm 3 Matching Knowledge Graph

Input: Knowledge graph KG; Symbols $\boldsymbol{Z}$; Rule set R
Output: new concept name c

$KG' = \mathrm{BFS}(KG, d)$ # Sub-graph depth d
$\phi = \mathrm{KGEmbed}(KG')$
$T = \mathrm{ConvertTriplet}(\boldsymbol{Z}, R)$
$E = \mathrm{Entity}(KG')$
$c = \mathrm{MaxScoreEntity}(E, \phi, T)$
return c

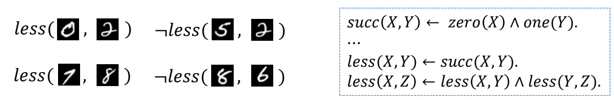


Figure 3: Dataset and knowledge in *Less-Than with New Digits* experiment.

ABL$_{nc}$ then trains a knowledge graph embedding model ϕ on the KG', which turns entities and relations into low-dimensional vectors. The embedding model ϕ can score a given triplet (h, r, t), where the higher score $\phi(h, r, t)$, the higher probability that the triplet holds.

The next step is converting labels and rules of new concepts into a set of triplets T. The basic idea is converting literals in learned rules R to KG triplets. For example, in the experiment of *Chess with New Pieces* (cf. Section 5.2), the rule $attack(V1, P, V2) \leftarrow new(P) \wedge diag(V1, V2)$ is converted to triplet $(new, \texttt{UsedFor}, moving\_diagonally)$.

As a result, ABL$_{nc}$ finds an entity in the sub-graph KG' that has the maximum score. This process is formalized as:

$$\max_{e \in E} \sum_{(h,r,t) \in T} \phi(e, r, t), \tag{7}$$

where E is the set of entities in the KG', and (e, r, t) is the triplet that replaces the new concept symbol h in the original triplet by an entity $e \in E$. An entity c with a maximum score of replaced triplets is returned.

5 Experiments

This section presents the experimental results on three neuro-symbolic tasks, to demonstrate that ABL$_{nc}$ could handle new concepts with improved model performance, and match the corresponding entities in the external knowledge graph. All experiments are repeated ten times on a server with Intel Xeon Gold 6242R CPU and Nvidia RTX 3090 GPU. The hyperparameters of ABL$_{nc}$ are determined by cross-validation on training data. The code is available for download[1].

[1] https://github.com/AbductiveLearning/ABL_nc

Task	Less-Than with New Digits		Chess with New Pieces		Multiples of Three	
	Perception Acc.	Reasoning Acc.	Perception Acc.	Reasoning Acc.	Perception Acc.	Reasoning Acc.
CNN	0.889±0.006	0.915±0.016	0.831±0.001	0.558±0.029	0.903±0.008	0.601±0.019
CNN_{new}	0.946±0.014	0.876±0.013	0.924±0.014	0.910±0.006	0.903±0.008	0.323±0.012
ABL	0.959±0.015	0.904±0.009	0.930±0.015	0.937±0.009	0.938±0.005	0.639±0.011
ABL_{nc}	**0.975±0.005**	**0.986±0.005**	**0.990±0.006**	**0.984±0.005**	**0.960±0.008**	**0.963±0.004**

Table 1: Test accuracy on perception and reasoning of different methods.

New Concept	Less-Than with New Digits					Chess with New Pieces			Multiples of Three
	zero	one	two	...	nine	bishop	king	queen	multiples of three
Entity	0	1	2	...	9	$chess\_bishop$	$chess\_king$	-	$multiple\_of\_3$

Table 2: Matched entities of new concepts in each task.

5.1 Less-Than with New Digits

Dataset. This is a novel dataset proposed in this paper for studying new concepts at perception level, as shown in Figure 3. An input example is a pair of handwritten digits, along with weak supervision that indicates whether the first digit is less than the second one. The dataset consists of 10k pairs of images, where the digits are randomly generated and their images are randomly sampled from the training set of MNIST. We randomly select a digit from 0 - 9 as the emerging new concepts at perception level, and report their average performance.

Perception and Reasoning Model. Before new concepts emerge, the initial machine learning model has been trained by known concepts, i.e., it could recognize digits $0-9$ except the new concept. We employ a convolutional neural network (CNN) as the perception model. Local Outlier Factor (LOF) (Breunig et al. 2000) is used as the new class detector. The knowledge base only contains domain knowledge of known concepts, implemented by an answer set program (ASP) (Lifschitz 2002). The right part of Figure 3 gives some sample rules. We use ILASP (Inductive Learning of Answer Set Programs) (Law, Russo, and Broda 2015) as the ILP system, and use a knowledge graph embedding model (Liu et al. 2022) based on Transformer.

Compared Methods. We compare ABL_{nc} with three baselines: 1) **CNN**: The initial model that can only recognize known digits and contains only knowledge of known concepts, serving as a baseline in our experiment. 2) $\textbf{CNN}_{\textbf{new}}$: A model where the perception model is updated on the new class detector's prediction, while the knowledge base remains the same. 3) **ABL** (Dai et al. 2019): Abductive learning where the perception model is first updated on the new class detector's prediction, and conducts abductive learning based on the initial knowledge base without adding knowledge of new concepts. We do not compare other hybrid neuro-symbolic systems like DeepProbLog (Manhaeve et al. 2018) and NGS (Li et al. 2020) since they assume no new concepts would appear in learning stage and would exhibit similar performance as ABL.

Results. The accuracy of perception and reasoning are shown in Table 1. The CNN baseline has the lowest accuracy, indicating that the emerging new concepts deteriorate the model performance. Although CNN_{new} has a higher perception accuracy than CNN by leveraging the prediction of new class detector, the reasoning accuracy is even inferior to CNN, probably due to the noise in new class detector. The ABL method performs better than CNN_{new} because abduction would revise the misclassified examples, but it still suffers from lack of new concept knowledge. ABL_{nc} significantly outperforms others by perception and reasoning ability because it could update perception and reasoning model simultaneously to handle the new concepts.

The learned rules after convergence are:

$$succ(X,Y) \leftarrow seven(X) \land new(Y). \tag{8}$$

$$succ(X,Y) \leftarrow new(X) \land nine(Y). \tag{9}$$

It is obvious that ABL_{nc} has successfully learned correct rules, where Rule (8) means the new concept (digit 8) is a succeeding number of seven, and Rule (9) means nine is a succeeding number of the new digit. There is no conflict between learned rules and knowledge base, and therefore the learned rules could directly be merged into KB.

We try to discover what the new concepts represent from ConceptNet (Speer, Chin, and Havasi 2017), a commonsense KG containing 34 million edges, and results are shown in Table 2. Learned rules such as (8) are converted into triplets like $(new, \texttt{IsA}, higher\_number\_than\_7)$. ABL_{nc} successfully matches all the new concepts with corresponding entities, which provides the end-user with semantic meanings and helps the rules become more interpretable.

5.2 Chess with New Pieces

Dataset and Setting. The *Chess* dataset is a more challenging one, which comes from the *extended n-queens* task in ABL (Dai et al. 2019) and contains 10k input examples, as shown in Figure 4. The inputs are images of randomly generated chessboards that contain several chess pieces (queen, king, bishop, knight, pawn, rook) and the associated labels are the validity of each board, where pieces are represented

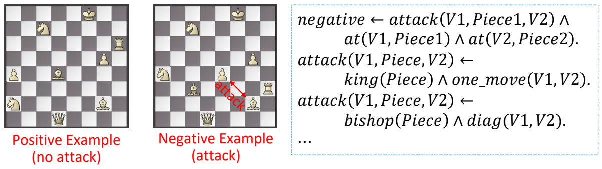


Figure 4: Dataset and knowledge in *Chess with New Pieces* experiment.

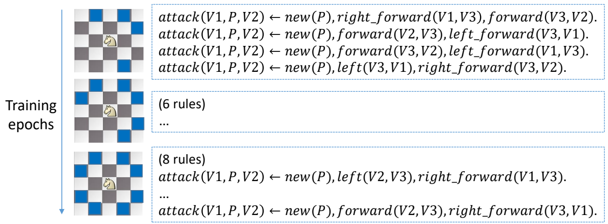


Figure 5: Learned rules during training in *Chess with New Pieces* experiment.

by randomly sampled MNIST images. The knowledge base includes rules about chess piece moves of known concepts. We choose *knight* as the new concept at perception level since its moves are the most complicated. Other experimental setups are the same as the first task.

Results. The results are shown in the middle of Table 1. With the emerging of new concepts, the performance of reasoning model drops to nearly 0.5, indicating the difficulty of this task. ABL is superior to CNN_{new} because although the given knowledge base lacks knowledge of new concepts, the perception model still benefits from it by abduction of known concepts. After updating machine learning model and reasoning model simultaneously, the proposed ABL_{nc} converges and achieves higher accuracy than other methods.

Figure 5 illustrates the learned rules of new concept knight as training goes on. At first, ABL_{nc} only learns four rules about which position the knight can move to, because of the low perception accuracy of new concept when it first appears. However, by abducing based on an incomplete knowledge base, the perception accuracy of new concepts is gradually improving during training, which in turn helps the reasoning model learn more complete rules. When the training converges, ABL_{nc} successfully learns all possible moves of the new concepts knight.

As shown in Table 2, ABL_{nc} successfully matches the corresponding "$chess\_king$" and "$chess\_bishop$" in KG for new concepts, and fails to match the remaining new concepts such as queen. We find that the conversion of new concepts to triplets plays an important role. ABL_{nc} would fail to match if the entities of converted triplets do not exist in KG. For example, if the converted triplet is $(new, \texttt{UsedFor}, moving\_in\_a\_straight\_line)$, ABL_{nc} would fail, while if the triplet becomes $(new, \texttt{UsedFor}, moving\_in\_direction)$, ABL_{nc} would succeed. It remains an open problem on how to effectively convert rules to triplets, we regard it as future work.

5.3 Multiples of Three

Dataset and Setting. The *Multiples of Three* dataset is proposed to test the ability to handle conflicts of new concepts rules in neuro-symbolic systems, where the new concepts emerge at reasoning level. It contains 10k examples, as shown in Figure 1 and case 1 of Figure 2, where the inputs include a list of single-digit MNIST numbers and the associated labels $\boldsymbol{z}$. The labels $\boldsymbol{z}$ include odd, $even$ and the new concept "$multiples\_of\_three$", as described in Section 4.3. As usual, labels of each digit are unknown. The initial knowledge base contains definitions of odd, $even$, and the initial perception model has been trained by 5% labeled MNIST images. We are required to learn the definition of new concepts and improve perception model. Different from previous experiments, since new concepts emerge at reasoning level, CNN_{new} means using pseudo-labels of digits for learning rules of new concepts, and adding them directly to knowledge base. Other experimental setups remain the same as the previous experiments.

Results. The accuracy of all models is shown in Table 1. Both the perception accuracy of ABL and ABL_{nc} improve compared with CNN, showing the benefit of abduction. However, the lack of knowledge of new concepts results in inferior perception and reasoning accuracy in ABL. CNN_{new} could learn part of the definition of new concepts, but the learned rules conflict with knowledge base and therefore dramatically degrade the reasoning performance. ABL_{nc} not only achieves the best perception and reasoning performance, but also refines the knowledge base by conflict resolution. Besides, we investigate the learned rule of ABL_{nc}, and find that it has learned the correct rules as Eq. (3). After conflict resolution, the rules of known concepts $even$ and odd are indeed refined as Eq. (5) - (6), leading to a knowledge base without conflict rules.

Table 2 shows the matched entity in an augmented ConceptNet (details provided in appendix). The new concepts is successfully matched to "$multiple\_of\_3$" in KG, providing a comprehensive understanding for human.

6 Conclusion

In this paper, we propose an approach to enable the Abductive learning paradigm to have the ability of knowledge refinement/enhancement. In detail, we propose ABL_{nc} (ABductive Learning with New Concept), in which the knowledge base is refined by conflict resolution of the learned rules of new concepts, while perception model is updated by abductive learning based on the augmented knowledge base. Moreover, the new concepts are matched with proper entities in knowledge graph. Experimental results on three tasks demonstrate that ABL_{nc} could adapt to emerging new concepts, leading to higher model performance and better human understanding. ABL_{nc} is a general-purposed approach with sufficient flexibility in implementation, e.g., the machine learning and rule learning ingredient can be replaced by other techniques. In this work, we assume that the new concepts and known concepts belong to the same category, and how to handle new concepts from different categories is an interesting future issue.

Acknowledgments

This research was supported by NKRDP (2020AAA0109401), NSFC (62176117, 61921006).

References

Ai, L.; Muggleton, S. H.; Hocquette, C.; Gromowski, M.; and Schmid, U. 2021. Beneficial and harmful explanatory machine learning. *Machine Learning*, 110(4): 695–721.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: Identifying density-based local outliers. In *SIGMOD Record*, volume 29, 93–104.

Cai, L.-W.; Dai, W.-Z.; Huang, Y.-X.; Li, Y.-F.; Muggleton, S.; and Jiang, Y. 2021. Abductive Learning with Ground Knowledge Base. In *IJCAI*, 1815–1821.

Cropper, A.; and Morel, R. 2021. Learning programs by learning from failures. *Machine Learning*, 110(4): 801–856.

Dai, W.-Z.; and Muggleton, S. H. 2021. Abductive Knowledge Induction from Raw Data. In *IJCAI*, 1845–1851.

Dai, W.-Z.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging machine learning and logical reasoning by abductive learning. In *NeurIPS*, 2811–2822.

De Raedt, L.; and Kimmig, A. 2015. Probabilistic (logic) programming concepts. *Machine Learning*, 100(1): 5–47.

Esposito, F.; Semeraro, G.; Fanizzi, N.; and Ferilli, S. 2000. Multistrategy theory revision: Induction and abduction in inthelex. *Machine Learning*, 38(1): 133–156.

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

Huang, Y.-X.; Dai, W.-Z.; Cai, L.-W.; Muggleton, S. H.; and Jiang, Y. 2021. Fast Abductive Learning by Similarity-based Consistency Optimization. In *NeurIPS*, 26574–26584.

Huang, Y.-X.; Dai, W.-Z.; Yang, J.; Cai, L.-W.; Cheng, S.; Huang, R.; Li, Y.-F.; and Zhou, Z.-H. 2020. Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing. In *ICDM*, 1070–1075.

Koller, D.; Friedman, N.; Džeroski, S.; Sutton, C.; McCallum, A.; Pfeffer, A.; Abbeel, P.; Wong, M.-F.; Heckerman, D.; Meek, C.; et al. 2007. *Introduction to statistical relational learning*. The MIT Press.

Law, M.; Russo, A.; and Broda, K. 2015. The ILASP system for learning Answer Set Programs. www.ilasp.com. Accessed: 2023-01-01.

Li, Q.; Huang, S.; Hong, Y.; Chen, Y.; Wu, Y. N.; and Zhu, S.-C. 2020. Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning. In *ICML*, 5884–5894. PMLR.

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

Lin, D.; Dechter, E.; Ellis, K.; Tenenbaum, J.; and Muggleton, S. 2014. Bias reformulation for one-shot function induction. In *ECAI*, 525–530.

Liu, Y.; Sun, Z.; Li, G.; and Hu, W. 2022. I Know What You Do Not Know: Knowledge Graph Embedding via Codistillation Learning. In *CIKM*, 1329–1338.

Ma, J.; and Perkins, S. 2003. Time-series novelty detection using one-class support vector machines. In *IJCNN*, volume 3, 1741–1745. IEEE.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, 3749–3759.

Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A. D.; and van de Weijer, J. 2020. Class-incremental learning: survey and performance evaluation on image classification. arXiv:2010.15277.

Mu, X.; Ting, K. M.; and Zhou, Z.-H. 2017. Classification under streaming emerging new classes: A solution using completely-random trees. *IEEE Transactions on Knowledge and Data Engineering*, 29(8): 1605–1618.

Muggleton, S.; and De Raedt, L. 1994. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19: 629–679.

Muggleton, S. H.; Lin, D.; and Tamaddoni-Nezhad, A. 2015. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. *Machine Learning*, 100(1): 49–73.

Raedt, L. D.; Dumancic, S.; Manhaeve, R.; and Marra, G. 2020. From Statistical Relational to Neuro-Symbolic Artificial Intelligence. In *IJCAI*, 4943–4950.

Raedt, L. D.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2): 1–189.

Schreiber, G.; and Raimond, Y. 2014. *RDF 1.1 Primer*. W3C Working Group Note. World-Wide Web Consortium.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 4444–4451.

Srinivasan, A. 2001. The ALEPH manual. Technical report, Machine Learning at the Computing Laboratory, Oxford University.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743.

Wrobel, S. 1996. First order theory refinement. *Advances in inductive logic programming*, 32: 14–33.

Zhang, Y.-J.; Zhao, P.; Ma, L.; and Zhou, Z.-H. 2020. An Unbiased Risk Estimator for Learning with Augmented Classes. In *NeurIPS*, 10247–10258.

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

Zhou, Z.-H. 2022. Open-environment machine learning. *National Science Review*, 9(8): nwac123.

Zhou, Z.-H.; and Chen, Z.-Q. 2002. Hybrid decision tree. *Knowledge-based systems*, 15(8): 515–528.

Zhou, Z.-H.; and Huang, Y.-X. 2022. Abductive Learning. In Hitzler, P.; and Sarker, M. K., eds., *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 353–369. Amsterdam: IOS Press.