SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2023, Vol. 66 122103:1–122103:13
https://doi.org/10.1007/s11432-020-3569-0

Abductive subconcept learning

Zhongyi HAN[1], Le-Wen CAI[2], Wang-Zhou DAI[3], Yu-Xuan HUANG[2], Benzheng WEI[4], Wei WANG[2*] & Yilong YIN[1*]

[1]*School of Software, Shandong University, Jinan* 250101*, China;*
[2]*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023*, China;*
[3]*Department of Computing, Imperial College London, London* SW7 2AZ*, UK;*
[4]*Center for Medical Artificial Intelligence, Shandong University of Traditional Chinese Medicine, Qingdao* 266112*, China*

Received 19 December 2020/Revised 6 January 2022/Accepted 25 June 2022/Published online 12 January 2023

Abstract Bridging neural network learning and symbolic reasoning is crucial for strong AI. Few pioneering studies have made some progress on logical reasoning tasks that require partitioned inputs of instances (e.g., sequential data), from which a final concept is formed based on the complex (perhaps logical) relationships between them. However, they cannot apply to low-level cognitive tasks that require unpartitioned inputs (e.g., raw images), such as object recognition and text classification. In this paper, we propose abductive subconcept learning (ASL) to bridge neural network learning and symbolic reasoning on unsegmented image classification tasks. ASL uses deep learning and abductive logical reasoning to jointly learn subconcept perception and secondary reasoning. Specifically, it first employs meta-interpretive learning (MIL) to induce first-order logical hypotheses capturing the relationships between the high-level subconcepts that account for the target concept. Then, it uses the groundings of the logical hypotheses as labels to train a deep learning model for identifying the subconcepts from unpartitioned data. ASL jointly trains the deep learning model and learns the MIL theory by minimizing the inconsistency between their grounded outputs. Experimental results show that ASL successfully integrates machine learning and logical reasoning with accurate and interpretable results in several object recognition tasks.

Keywords abductive learning, subconcept learning, logical reasoning, subconcept set selection, meta-interpretive learning

Citation Han Z Y, Cai L-W, Dai W-Z, et al. Abductive subconcept learning. Sci China Inf Sci, 2023, 66(2): 122103, https://doi.org/10.1007/s11432-020-3569-0

1 Introduction

Bridging deep learning and logical reasoning is hugely significant in the future of artificial intelligence (AI) [1]. Deep learning has been widely applied and has achieved human-compatible or even above-human performance in many fields. For example, the performance of deep neural networks (DNN) is on par with humans on tasks like image classification [2]. Most of these approaches adopt an end-to-end paradigm for learning; i.e., they learn discriminative models directly mapping from low-level sensory information into high-level semantically meaningful labels. However, this formulation has several deficiencies, e.g., it makes the learned models (1) hardly explainable [3]; (2) requires a large amount of training examples [4]; (3) hardly performs well or even becomes useless if the environment changes [5]. Logical reasoning performs well in high-level symbolic reasoning with high interpretability and robustness, but it can only process symbolic data. Therefore, how bridging deep learning and logical reasoning has been deemed as the holy grail challenge for the AI community [6].

To achieve this goal, Zhou [6] and Dai et al. [7] proposed the pioneering framework abductive learning (ABL). ABL consists of a machine learning model for interpreting inputs into primitive logical facts and a logical model for reasoning out the final result based on first-order logical background knowledge. In practice, ABL performs well to resolve hand-written equation decipherment puzzles. The inputs of

* Corresponding author (email: wangw@lamda.nju.edu.cn, ylyin@sdu.edu.cn)

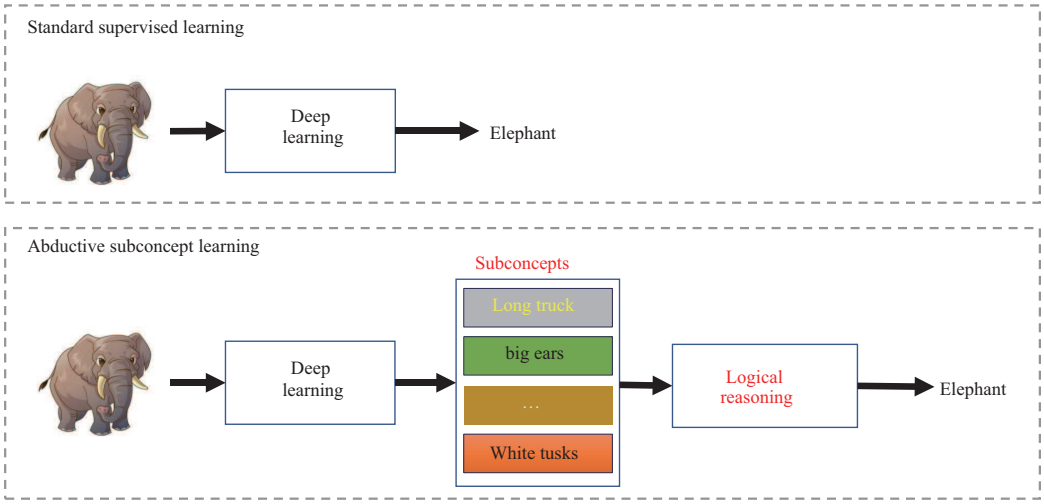

Figure 1 (Color online) An illustration of abductive subconcept learning. Compared with standard supervised learning, abductive subconcept learning is able to learn semantic subconcepts for improving the comprehensibility and generalization of deep learning models.

ABL have already been pre-processed into sequences of subcomponents. Moreover, Manhaeve et al. [8] proposed the DeepProbLog to combine DNNs and probabilistic logic (ProbLog [9]). DeepProbLog applies to human reasoning tasks with partitioned inputs, e.g., sorting a sequence of digital images.

While pioneering studies have achieved great success, the proposed approaches can only use for high-level reasoning tasks. They still cannot process unpartitioned data where the partitioned subcomponents are unavailable. At present, unpartitioned data exists widely in the low-level cognitive tasks that are actual application scenarios of AI, e.g., classification, clustering, and ranking. Accordingly, this paper aims to study unifying neural network learning and logical reasoning on unsegmented image classification tasks, which is still an open problem.

We propose to exploit subordinate concepts (subconcept) for bridging neural network learning and logical reasoning on unsegmented image classification tasks. Generally speaking, we assume that each target class consists of several subcomponents from human knowledge. Moreover, we view each subcomponent as an individual subconcept. The target concept to be learned is constructed by subconcepts or some complex relationships between them. As illustrated in Figure 1, the concept of the elephant consists of several subconcepts, such as long trunks, big ears, and white tusks. We can identify these subconcepts from a single input image and perform secondary reasoning about their relationships to deduce the holistic label (i.e., elephant).

In this paper, we propose an abductive subconcept learning (ASL) approach to exploit and learn the subconcepts. Given a learning task (e.g., elephant recognition), ASL firstly uses meta-interpretive learning to induce the logical hypothesis of the elephant class that can decompose into a high-level subconcept set. ASL then uses DNNs to identify the subconcepts of unpartitioned data and then uses logical reasoning to perform secondary reasoning on subconcepts to infer the class label. The learning process is coordinated by minimizing the inconsistency between logical hypotheses and perceptual DNNs. A series of empirical studies on object recognition tasks demonstrate that ASL can (1) discover high-level subconcepts to interpret the recognition results, (2) outperform deep learning models with fewer data, and (3) be reused in new tasks of different environments that share some subconcepts with the source task.

The core contributions include the following.

• Different from previous studies, we introduce subconcepts to bridge neural network learning and logical reasoning in real-world applications with unpartitioned data towards achieving more explainable and high-level deep learning.

• We propose the abductive subconcept learning approach to simultaneously achieve subconcept set selection and subconcept learning by integrating meta-interpretative, abductive, and deep learning.

• To verify this new approach, we construct various datasets and design sufficient experiments. A series of empirical studies have demonstrated its robustness and effectiveness.

2 Related work

Modeling the secondary reasoning process has been studied for many years in machine learning. Generally speaking, the hierarchical information processing paradigm of DNNs and capsule neural networks can be regarded as a particular form of secondary reasoning [10,11]. The raw inputs are processed layer-by-layer and eventually form high-level features that could be semantically meaningful [2].

Capsule neural networks involve similar ideas in deep learning [11]. They model hierarchical relationships inside the knowledge representation of a neural network capsule-by-capsule. The knowledge representation has various properties of a particular entity that are more explainable and meaningful than convolutional neural networks. The stacking technique also can be viewed as a secondary reasoning process [12,13]. Stacking is an ensemble technique widely used in statistical machine learning. The multiple different learners build intermediate predictions that can be regarded as subconcepts. The final model stacked on top of the others performs a similar function as secondary reasoning based on the intermediate subconcepts. However, the learned hierarchical information of the methods mentioned above is vague, while the learned subconcepts of ASL are so explicit that they have interpretability.

The problem formulation of ASL is related to multi-instance learning [14]. The setting of multi-instance learning is that an example can be described by an existing bag composed of instances, in which one instance can be considered a subconcept [15]. The label space of multiple instance learning is also similar to ASL in that the holistic label is binary. The main difference is that the instances of an example are presumed to be existed and partitioned well in the first place, such that multi-instance learning is subject to feature or input space [16]. Accordingly, the problem formulation of ASL can degenerate into a multi-instance learning problem. To further verify the effectiveness of ASL, we implemented ASL under the multi-instance learning setting with multiple benchmarks in Section 5.

Multi-label multi-instance learning (MIML) extends the multi-instance learning into a more generalized scenario where each training example is associated with multiple instances and multiple class labels. Thus MIML deals with data objects that are represented by a bag of instances and associated with a set of class labels simultaneously [17]. The "subconcepts" in ASL are almost equivalent to the labels of "instances" in the MIML setting. The target concept of ASL consists of a set of subconcepts, and an example may contain some irrelevant subconcepts. In the MIML setting, an example is a bag of instances, and the holistic label of an example is determined by the set of instances and their relations within this bag. The main differences between ASL and MIML are three-fold. First, MIML assumes that each example of the target concept is already partitioned into a set of subconcepts. The subconcepts of an example are presumed to be existed and partitioned well in the first place. Second, most MIML studies assume that if a bag contains a positive instance, then the bag is positive [18,19]. In other words, most MIML studies assume that there are no direct logical relationships between instances. Finally, MIML can easily handle the multi-label learning problem, which is very complex and challenging for ASL. Pioneering MIML studies can provide a trustworthy theoretical foundation for the profound analysis of ASL. Wang et al. [20] innovatively demonstrated that the MIML hypothesis class constructed from a multi-instance single-label hypothesis class is PAC-learnable and proved the generalization bound for the MIML problem. Zhou et al. [19,21] innovatively proposed the SUBCOD algorithm to transform the multi-instance single-label task as a MIML problem. SUBCOD first clusters all instances, treats each cluster as a subconcept, then uses a classifier mapping the derived subconcepts to the original single labels. The clusters process is similar to a subconcept selection process, and the mapping process is similar to a secondary reasoning process.

Disentangled representation learning techniques separate each feature into narrowly defined variables and encode them as separate dimensions, which can be regarded as a subconcept selection problem [22,23]. However, unsupervised learning of disentangled representations has been demonstrated that it is fundamentally impossible without inductive biases [24]. Several studies attempt to utilize neural connection mechanisms to achieve logic-like secondary reasoning [25, 26]. For example, PrediNet uses an attention mechanism to learn propositional and relational representations by integrating cascade neural networks [26].

Probabilistic logic program [27] and statistical relational learning [28] aim at integrating probabilistic inference and logical reasoning. However, they usually require semantic-level inputs. Neural logic machine [25] attempts to use neural connection mechanisms to achieve logic-like secondary reasoning, but still has the drawbacks of deep learning, as mentioned in Section 1.

Abductive learning proposed by [6, 7] leverages logic programming towards bridging machine learning

and logical reasoning to achieve more explainable secondary reasoning. Inspired by the human abductive problem-solving process, it can simultaneously optimize the machine learning and logical reasoning models using weak annotations and domain knowledge written as first-order logic rules. Like abductive learning, ASL brings in abductive logic reasoning and background knowledge as a human-like reasoning process. In addition, ASL advances abductive learning into more common real-world applications through two main improvements. Firstly, ASL can induce logical hypotheses of target concepts to discover subconcepts. Secondly, ASL can identify subconcepts that are always not partitioned from unpartitioned data.

3 Preliminaries

This section presents the pioneering abductive learning framework and meta-interpretive learning.

3.1 Abductive learning

The input of ABL is a logical sequence of examples $X_i = (x_1^i, x_2^i, \ldots, x_n^i)$. The logical sequence refers to the sequence's inner examples that have strict logical relationships. Note that the ground-truth label of each example x_n^i is unavailable. The output is a holistic label $Y_i \in \{0, 1\}$ to justify the correctness of the logical sequence.

ABL connects a machine learning module with an abductive logical reasoning module and bridges them with consistency optimization [7]. The objective of ABL is to learn a hypothesis that is consistent with background knowledge and training examples. In short, ABL works as follows. The machine learning module is used to obtain each example's pseudo-label in an input sequence. The logical reasoning module treats pseudo-labels as groundings of primitive concepts to infer the holistic label of the sequence, such as true or false. Suppose the holistic label is different from the ground-truth label. In that case, a consistency optimization module is called to revise the pseudo-labels of each example, which are then used for retraining the machine learning model. In summary, ABL consists of three critical modules as follows.

• Machine learning module. Given a sequence of examples, this module can predict a counterpart sequence of pseudo-labels, which can be considered groundings of possible primitive concepts. If the pseudo-labels contain mistakes, this module will be re-trained using 'ground-truth' labels that are the revised pseudo-labels returned by logical abduction.

• Logical abduction module. It is the logical formalization of abductive reasoning inspired by the abductive logic programming [29]. Based on primitive pseudo-labels as observed facts and background knowledge expressed as first-order logical clauses, logical abduction can abduce ground hypothesis, which is a possible explanation for the observed facts. Another role of logical abduction is to infer the holistic label for the input pseudo-labels.

• Optimization. In the setting of ABL, we only know the holistic ground-truth label, that is, the holistic label of a logical sequence data, but do not know the ground-truth label of each example. ABL attempts to maximize the consistency between pseudo-labels of examples and background knowledge. When the machine learning model is not convergent, ABL needs to correct the pseudo-labels to achieve consistent abductions. ABL uses a heuristic function to estimate which pseudo-labels are misperceived according to the holistic ground-truth label and background knowledge. ABL then applies logical abduction to abduce the possible correct pseudo-labels as 'ground-truth' for re-training the machine learning model. Since this optimization objective is non-convex, ABL solves it by utilizing a derivative-free optimization tool RACOS proposed by [30]. We follow this optimization process using the logical abduction approach.

3.2 Meta-interpretive learning (MIL)

MIL is an inductive logic programming system. It supports predicate invention and efficient learning of logical hypotheses because MIL can execute high-order logic programming [31]. The inputs of MIL include a knowledge base KB and a set of logical facts E. KB consists of manually designed domain knowledge for improving the hypothesis induction. E is composed of a few positive examples and negative examples, i.e., $E = E^+ \cup E^-$. The task is to learn a hypothesis H that defines the target concept class satisfying $B \wedge H \models E$, where $B = \mathrm{KB} \cup M$. M is a set of meta-rules. Meta-rules are second-order logic clauses that view the predicates and functions of first-order logic (FOL) as variables. These variables can be grounded by abductive reasoning from B and E. The symbol $\models$ stands for entailment, which represents

that the label of E is correct only if both B and H are satisfied. To learn logical hypotheses, MIL uses second-order abduction to convert inductive problem to deduction problem $B, \neg E \models \neg H$, where $\neg H$ is the negation of H such that the raw hypothesis can get from the negation of inverse entailment result. A logical hypothesis H of a concept is composed of a set of clauses,

$$A \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge B_n\,, \tag{1}$$

where A is an atom. An atom is a formula with no deeper propositional structure, that is, a formula that contains no logical connectives ($\vee$, $\wedge$) or equivalently a formula that has no strict subformulas. B_i is literal. A clause without any variable is grounded, and a grounded atom is called ground fact.

The workflow of a MIL is to continuously prove a set of ground facts according to background knowledge by fetching higher-order meta-rules. The proving process is a predicate substitution process, and a predicate is invented if the substituted predicates do not exist in the knowledge base.

4 Abductive subconcept learning

4.1 Learning set-up

We first consider the familiar supervised learning setting where the learner receives a sample of m labeled training examples $\{(x_i, y_i)\}_{i=1}^m$ drawn from a joint distribution $\mathcal{D}$ defined on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input set and $\mathcal{Y}$ is the label set. Let $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denote a loss function defined over pairs of labels. Let $f : \mathcal{X} \to \mathcal{Y}$ denote a classifier. For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ and any classifier $f \in \mathcal{F}$, let $\epsilon_{\mathcal{D}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathcal{L}(f(x), y)$ denote the expected risk.

For abductive subconcept learning, a subconcept set $\mathcal{Z}$ connects an input set $\mathcal{X}$ and a holistic label set $\mathcal{Y}$. $\mathcal{Y} \in \{0, 1\}$ for binary classification tasks is the main analysis in this paper. Let $\mathcal{Z}_0$ denote the subconcept set of the negative class and $\mathcal{Z}_1$ denote the positive class. Let $z_i \in \mathcal{Z}$ denote the subconcepts of the i-th example and $z_i = \{z_{i,1}, z_{i,2}, \ldots, z_{i,n}\}$ where $z_{i,j}$ denotes a subconcept. Subconcepts refer to the high-level subcomponents that can constitute the target class concept (i.e., holistic label). For example, the subconcepts of long trunks, big ears, and white tusks can constitute the elephant concept. Subconcepts can be viewed as high-level characteristics of the target class to be recognized. Given an input x_i, the objective is to learn the hierarchical subconcepts z_i and use them to infer the holistic label $\hat{y}_i$. We assume that the subconcept set of each class exists in the real world. However, two challenges appear: (1) the subconcept set of each class is unknown, and (2) furthermore, the ground-truth subconcepts of examples are uncertain because the subconcepts of different examples from the same class are varied due to different views. For example, some elephant pictures may miss elephant trunks, which leads the ground truth of the trunk subconcept to be false. Moreover, we should assume that the subconcepts of examples are consistent with common senses; e.g., an artwork of an elephant with duck legs is unacceptable.

This subsection presents the abductive logical reasoning (ASL) approach. Recall that in the learning setup, we have mentioned two significant challenges: the subconcept set of each class is unknown, and the ground-truth subconcepts of training examples are uncertain because they are the subset of the target class's subconcept set. Accordingly, ASL has a subconcept set selection process to discover the subconcept set of each class to be recognized (see Subsection 4.2). ASL also has a subconcept learning process to learn the subconcepts of training examples without ground-truth annotations (see Subsection 4.3).

4.2 Subconcept set selection

Subconcept set selection aims to utilize logical induction for reasoning out a subconcept set for each class in specific tasks. Subconcept set selection consists of two steps: (1) inducing a logical hypothesis for each class, (2) decomposing each logical hypothesis into a subconcept set, respectively. In the first step, we utilize MIL to induce logical hypotheses. Given a learning task (e.g., elephant recognition), the user should provide a knowledge base KB and a set of logical facts E. The knowledge base includes some background knowledge, e.g., the logical relations between elephant body structures. The logical facts include a few positive examples (no more than ten) composed of the general characteristics of elephants. After running MIL, we would obtain a logical hypothesis of H that depicts the substantive characteristics of the target class, e.g., elephant $\leftarrow$ long truck $\wedge$ big ears $\wedge \cdots \wedge$ white tusks. A more logical reasoning example is presented in Subsection 4.4 for understanding this process better.

In the second step, since each logical hypothesis comprises a set of Horn clauses, we naturally view each literal of Horn clauses as a subconcept. Therefore, a subconcept set refers to a group of literals. The logical conjunctions capture the relationship between subconcepts to account for the target class. Accordingly, a logical hypothesis can be converted into

$$K \leftarrow \text{subconcept}_1 \wedge \text{subconcept}_2 \wedge \cdots \wedge \text{subconcept}_n, \tag{2}$$

where K denotes a class and the subconcept set $\mathcal{Z} = \{\text{subconcept}_1, \text{subconcept}_2, \ldots, \text{subconcept}_n\}$. The amount n of subconcepts is usually very small upon the specific task. Note that the subconcept set of each class can be discovered from the knowledge base and the logical hypotheses; however, the subconcepts of examples of each class are a subset of the subconcept set of the class. Therefore, the subconcept set can be viewed as meta information that supervises the machine learning model to learn and predict accurate subconcepts of each example. In such a way, the subconcept set selection is linked to perception and logical reasoning.

4.3 Subconcept learning

Subconcept prediction. Since the subconcepts of each instance are uncertain, we design a multiple-output deep learning model f to identify subconcepts of examples. The number of outputs is the same as the subconcept number of a class' subconcept set. Each output corresponds to a subconcept. Formally, given an input x_i, let O_i denote the outputs and $O_{i,j}$ denote the j-th output. $O_{i,j}$ represents the probability of a subconcept $z_{i,j}$, i.e., $P(z_{i,j}) = O_{i,j}$. If $O_{i,j} \geqslant \tau$, $z_{i,j} = 1$, otherwise $z_{i,j} = 0$. We set $\tau = 0.5$.

Holistic label prediction. We embed the predicated subconcepts z_i into the logical hypothesis of H to generate the grounded hypothesis clauses. A logical abduction module uses the grounded hypothesis clauses to infer the holistic label $\hat{y}_i$. Formally, the holistic label $\hat{y}_i$ is inferred by combining background knowledge BK, logical hypothesis H, and the input example x_i:

$$\text{BK} \wedge H \wedge z_i \Rightarrow \hat{y}_i, \quad \text{where} \quad x_i \wedge f \Rightarrow z_i. \tag{3}$$

Model optimization. The essence of abductive subconcept learning is to optimize the multiple-output deep learning model. To optimize it, we adopt the optimization module of abductive learning. When the deep learning model f is under-trained, the pseudo-labels of subconcepts z_i have errors with a large probability. The optimization module calls the logical abduction module to abduce possible true labels of subconcepts for re-training the deep learning model. The objective of the optimization module is to maximize the consistency $\text{Con}(z_i, y_i)$ of predicated subconcepts with the ground-truth holistic label y_i according to the BK, logical hypothesis (H), and the deep learning model f with parameters ($z_i = f(x_i; \theta)$). Based on (3), this objective can be formulated as

$$\arg\max_{z_i \in \mathcal{Z}} P(\text{Con}(z_i, y_i) | \text{BK}, H, f(x_i; \theta)), \tag{4}$$

where P is the probability of consistency, and the deep learning model is fixed when maximizing the consistency. When inconsistency occurs, according to the abductive learning [7], the optimization module tries to solve this problem by finding possibly correct subconcepts. It firstly substitutes some possibly incorrect subconcepts to blank variable "_". Intuitively, the abduction model finds possibly incorrect subconcepts by trial and error. It then lets the logical abduction module to abduce an optimal subconcept filling in "_" to ensure a maximal consistency by the heuristic search. After finding possibly correct subconcepts, the optimization module uses them to optimize the deep learning model by

$$\arg\min_{\theta} \mathbb{E}_{x_i \sim \mathcal{D}} \mathcal{L}(f(x_i; \theta), z_i), \tag{5}$$

where $\mathcal{D}$ is the underlying distribution and $\mathcal{L}$ is a binary cross-entropy loss function in practice. The optimization process completes until reaching the set number of iterations. Algorithm 1 elaborates the optimization and interaction processes between deep learning and MIL modules.

Figure 2 (Color online) An illustration of the generated arch dataset. The blocks have various shapes, sizes, and colors. The positive class is (a) arch while the negative class is (b) not arch.

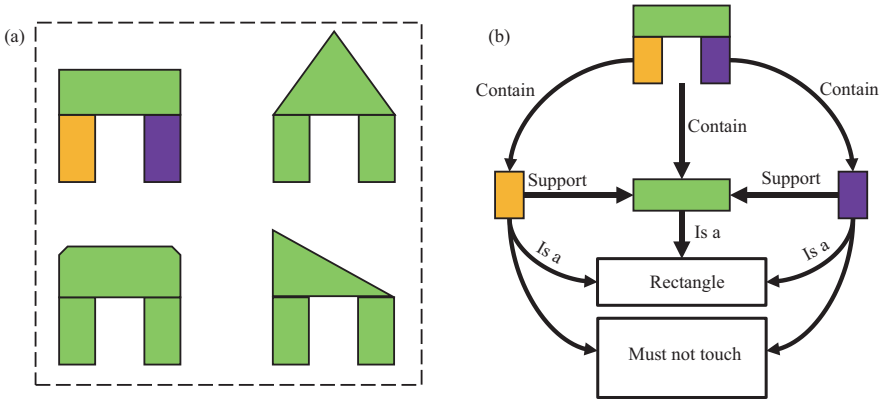

Figure 3 (Color online) An illustration of the concept of arch and its subconcepts with relationships. The arch concept is constructed by several high-level subconcepts, such as `support`, `untouch`, `triangle`, `rectangle`. (a) Arches; (b) subconcepts.

Algorithm 1 Abductive subconcept learning algorithm

Input: Knowledge base KB, logical facts E, training dataset D, epoch E;
Output: Deep model f, subconcept set $\mathcal{Z}$;
/* stage 1: subconcept set selection */
Running Metagol [31] to learn the logical hypothesis H of target class based on KB and E;
/* stage 2: subconcept learning */
Initialize multi-output deep model f;
for e to E **do**
  $\bar{D} = [\,]$;
  for $x_i \in D$ **do**
    $\tilde{z}_i = f(x_i)$;
    $\tilde{y}_i = \text{infer}(\tilde{z}_i, \text{KB}, H)$ by (3);
    if $\tilde{y}_i \neq y_i$ **then**
      $z_i = \text{abduce}(\tilde{z}_i, \text{KB}, H)$ by (4);
    else
      $z_i = \tilde{z}_i$;
    end if
    $\bar{D}$.append((x_i, z_i));
  end for
  Updating model f via $\bar{D}$;
end for

4.4 A running example

We let the arch image recognition task as a running example. As illustrated in Figure 2, the positive class is arch while the negative class is not arch. This task is similar to the relational learning example in [32]. As shown in Figure 3, the synthetic images are generated with triples of blocks, where the first two blocks are the sides of an arch, and the third block is the top. A part of background knowledge is shown as follows:

```
ako(stable_poly,triangle). % Triangle is a kind of (ako) stable polygon.
ako(stable_poly,rectangle). % Rectangle is a kind of stable polygon.
ako(unstable_poly,hexagon). % Hexagon is a kind of unstable polygon.
```

We used Metagol [31] to learn the logical hypothesis of the arch concept. The logical facts include several positive and negative examples. The induced logical hypothesis is shown as follows.

```
arch(A,B,C) :- support(A,C), support(B,C), \+touch(A,B), ako(stable_poly,C).
```

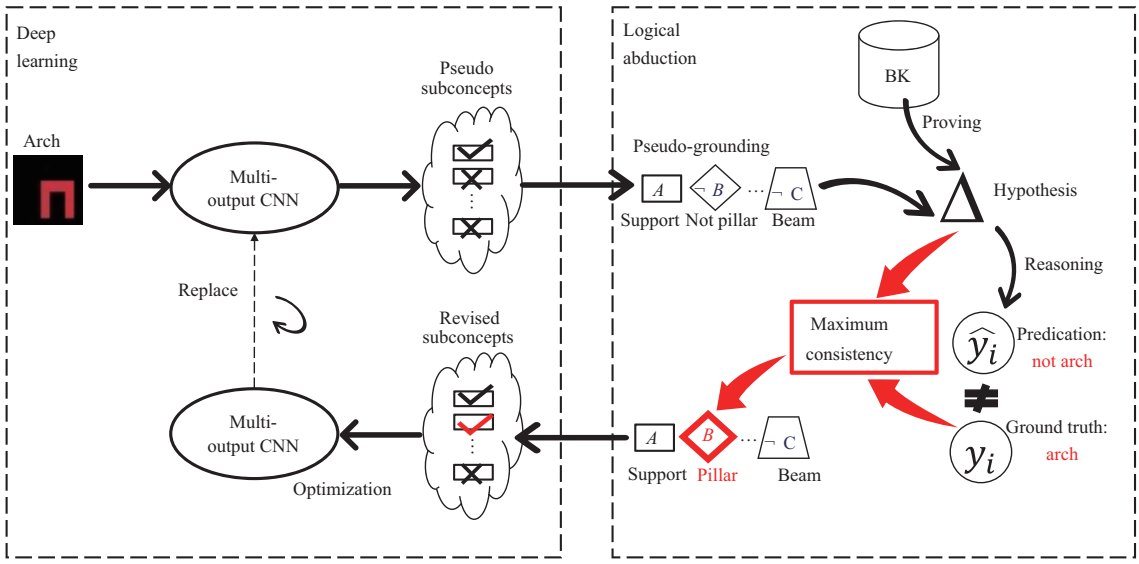

Figure 4 (Color online) The learning process of subconcepts from unpartitioned data. The interaction between deep learning and logical abduction can help each other obtain revised subconcepts. The deep learning model and logical abduction are tightly connected.

This hypothesis indicates that an arch consists of two pillars A and B, and a beam C, in which pillars A and B must support C. A does not touch B, and C is a stable polygon. Accordingly, the subconcept set of the arch is $\mathcal{Z} = \{\text{support}, \text{pillar-A}, \text{pillar-B}, \text{touch}, \text{beam-C}\}$. A five-output convolutional neural network and the logical abduction module are combined to identify subconcepts from input images. As shown in Figure 4, the interaction between the multi-output convolutional neural network and the logical abduction module can help each other to obtain revised subconcepts.

5 Experiments

Since the motivation of abductive subconcept learning is novel, it would be premature to apply the ASL framework to rich, complex data before we have a basic understanding of its properties and its behavior. Therefore, our experimental goals in this paper are (1) to test the primary hypothesis that the ASL framework can learn high-level subconcepts for unifying machine learning and logic reasoning, (2) to verify the primary hypothesis that the ASL framework can apply to real-world tasks and predict holistic labels accurately, and (3) to validate the ability that the learned ASL model is reusable to new tasks that share partial subconcepts with the source task. To do this, we first introduce implemented machine learning algorithms to provide a benchmark for further studies. We then evaluate the proposed approach on several low-level cognitive tasks against conventional deep learning models.

5.1 Setup

Arch recognition dataset is a newly-designed synthetic arch classification dataset. This dataset has 10000 training images, 1000 validation images, and 1000 testing images. Each example is a randomly generated image with a size of $32 \times 32 \times 3$. The blocks have various shapes, sizes, and colors, as illustrated in Figure 2. An advantage of synthetic datasets is that the ground-truth label of subconcepts of each example is available. Thus, this dataset can test the arch recognition accuracy and verify the recognition accuracy of subconcepts. This task could represent real-world tasks because it does not have a very complex relation between subconcepts and concept class. The used background knowledge and learned hypothesis are shown in Subsection 4.4.

Generalization dataset (GD) is a newly-designed dataset for the usual binary classification of positive and negative. For proving the universality of ASL, we constructed this dataset to mimic the general object classification. We let the set of subconcepts $\{\text{star}\ (S), \text{circle}\ (C), \text{triangle}\ (T), \text{rectangle}\ (R)\}$ as analogies of real-world subconcepts, respectively. Since the label of each subconcept is positive or negative, there are 16 combinations. We randomly chose three combinations as a positive class while the other

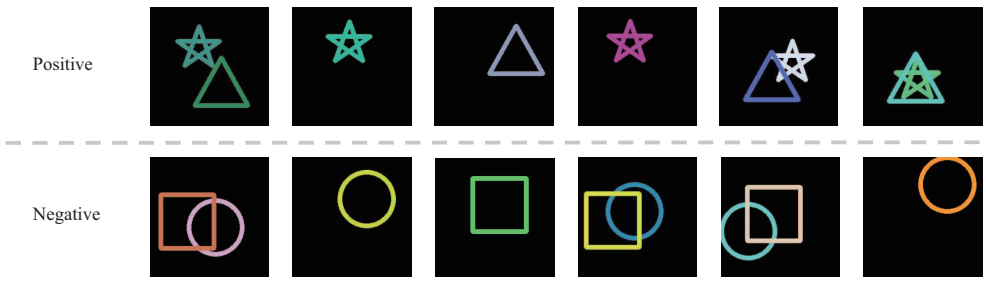

Figure 5 (Color online) An illustration of examples of the generalization dataset.

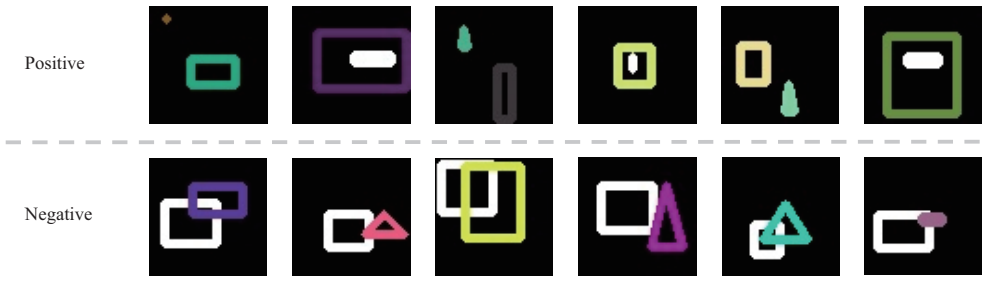

Figure 6 (Color online) An illustration of examples of the Bongard problem dataset.

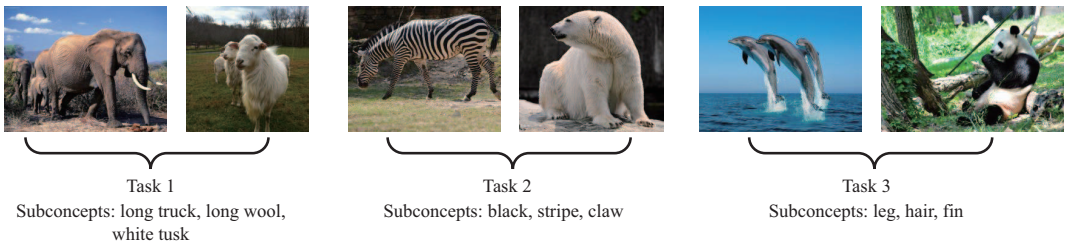

Figure 7 (Color online) We define three tasks based on the animal recognition dataset.

combinations were a negative class. Each image is randomly generated with a size of $32 \times 32 \times 3$, and the shapes have various sizes and colors, as shown in Figure 5.

Bongard problem dataset is a newly-designed dataset of reasoning tasks. The Bongard problem is a kind of puzzle. We constructed this dataset to verify the logical reasoning ability of ASL. As shown in Figure 6, we design three types of relationships: in, out, overlapping between two shapes in each image. The subconcept set of positive class is $\{\text{shape1}\ (S1), \text{shape2}\ (S2), \text{in}, \text{out}\}$, while the subconcept set of negative class is $\{\text{shape1}, \text{shape2}, \text{overlapping}\}$. This dataset also has 10000 training images, 1000 validation images, and 1000 testing images. Each image has a size of 32×32. The training and testing examples contain different shapes with different colors for proving the generalization of ASL. According to the setting, the logical hypotheses of the true and false concepts are

```
True(S1,S2):-in(S1,S2);out(S1,S2),
             not overlap(S1,S2).
False(S1,S2):-overlap(S1,S2),
             not in(S1,S2), not out(S1,S2).
```

Animal recognition dataset is a real dataset created using the dataset of animals with attributes (AwA) [33]. AwA dataset consists of 37322 images and 50 animal classes. As shown in Figure 7, we randomly chose six categories to construct three animal recognition tasks, in which each task consists of two animal classes. The objective of this dataset is to verify the learning ability of ASL in real-world applications. The discovered subconcept set of each category is presented in Figure 7.

We compare the proposed ASL approach with state-of-the-art methods: CNNs and PrediNet [26]. PrediNet is an attention-based network designed for the task of relation games. PrediNet consists of a CNN module connected by a multi-branch multi-head attention module followed by a fully connected layer. Since synthetic datasets' images are small, we design a shallow CNN as the backbone. The shallow

Table 1 Classification accuracy (%) on three synthetic datasets. Experiments were run five times, and an average of the classification accuracy (± a standard error of a mean, bold text represents the best result) is reported

Method	Arch	GD	Bongard
PrediNet	99.5 ± 0.2	98.5 ± 0.4	75.9 ± 3.2
CNN	99.6 ± 0.1	99.1 ± 0.3	80.3 ± 2.4
ASL	**99.9** ± 0.1	**99.9** ± 0.1	**85.8** ± 1.7

Table 2 Subconcept recognition accuracy (%) on three synthetic datasets (bold text represents the best result)

Method	Arch	GD	Bongard
Multiple CNNs	93.4 ± 2.4	**82.9** ± 3.9	85.5 ± 1.6
Multi-output CNN	**94.0** ± 2.2	78.9 ± 4.5	**90.5** ± 2.1

CNN comprises two convolution layers with a kernel of 5×5 and three fully connected layers. Batch normalization layers are embedded in each layer except for the last layer. For the animal recognition task, we use the residual neural network (ResNet-32) [34] as the backbone of ASL. The compared CNNs and the PrediNet' CNN module have the same structures as the backbone of ASL.

We implement our approach in Pytorch and Prolog. For a fair comparison, the parameters of CNNs are initialized from scratch using Kaiming uniform; i.e., all the models do not require pretraining. We use the mini-batch Adam optimizer. The learning rate is set to 0.0005. The training epoch is 100, and the batch size is set to 50. All the results are average from five repeated experiments.

5.2 Result

Table 1 reports the results on three synthetic datasets. The newly-proposed ASL approach significantly outperforms compared methods on the three tasks. ASL achieves comparable or above performance compared with CNN and PrediNet on the arch recognition task. ASL also achieves comparable performance on the generalization dataset, demonstrating its universality. ASL remarkably outperforms compared methods over 5% accuracy improvements on the Bongard puzzle problem, demonstrating that ASL can also apply in simple logical reasoning tasks. Figure 8 reports the results on the real dataset, where ASL makes a remarkable performance boost at the different number of training examples, which verifies that ASL can apply in actual application scenarios of AI. These great results verify our insight that ASL can successfully bridge deep learning and logical reasoning in low-level cognitive tasks. From another view, these results imply that our algorithm overcomes the two challenges of subconcept set selection and subconcept learning.

5.3 Discussion

This subsection presents four strengths of abductive subconcept learning compared to deep neural networks.

Interpretability. Essentially, subconcepts are the representative characteristics of data. These characteristics can explain the classification results thoroughly and make users understand the reason. On the other hand, the logical abduction module of ASL inherently has strong interpretability. For example, an image is classified into an arch class because the subconcept recognition results of the image are that pillar-A supports beam-C, pillar-B supports beam-C, pillar-A does not touch pillar-B, and beam-C is a rectangle. According to background knowledge, a rectangle is a stable polygon. The user can easily understand the classification result. Moreover, Table 2 reports the results of the subconcept recognition accuracy of multi-output CNN and multiple CNNs on three synthetic datasets. These results demonstrate that MIL can improve the subconcept recognition performance of ASL thanks to the interaction between the deep learning and MIL modules. Multi-output CNN refers to one output corresponding to a subconcept, while multiple CNNs refer to one CNN corresponding to one subconcept. Multi-output CNN and multiple CNNs have no significant difference, achieving promising performance. These results demonstrate that ASL can identify subconcepts from unpartitioned data, which is a guarantee of interpretability.

Data requirements. While Table 1 has demonstrated the strengths of ASL in performing secondary reasoning and object recognition, we provide a broader spectrum for more in-depth analysis. Figure 8 demonstrates that ASL can outperform deep learning models with fewer data, i.e., ASL requires fewer data. For example, in task 3, ASL uses 100 training examples, outperforming the performance of CNN

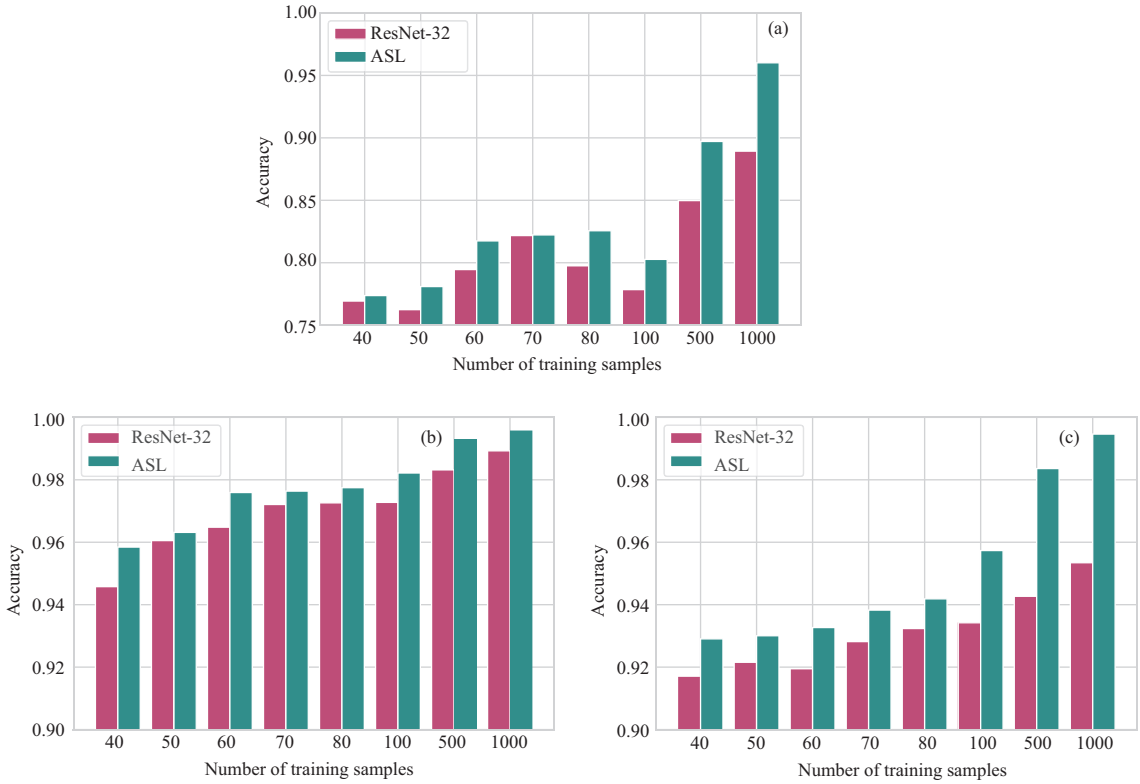

Figure 8 Classification accuracy of the animal recognition dataset with various numbers of training examples. (a) Task 1; (b) task 2; (c) task 3.

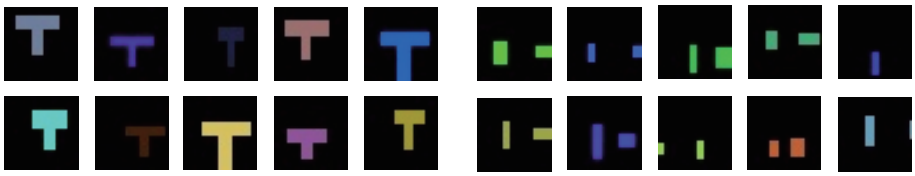

Figure 9 (Color online) An illustration of the generated hammer dataset. (a) Hammer; (b) not hammer.

with 1000 training examples. This result indicates that using background knowledge may reduce the data requirements such that we should make use of the advantages of logical reasoning.

Model reuse. We defined a hammer recognition task and constructed a hammer recognition dataset to validate ASL's reusability. This dataset has 1000 testing images, and the representative examples are shown in Figure 9. The subconcept set of the hammer concept is $\{\text{rectangle-A}, \text{rectangle-B}, \text{support}\}$, where rectangle-A must support rectangle-B. We reused the deep learning models of ASL that can recognize the subconcepts of rectangle and support in the arch classification tack. We implemented them directly on the hammer recognition dataset without any re-training. We also transferred the trained CNN model of arch recognition to this dataset. As a result, our algorithm achieves 88.52% accuracy and 95.51% subconcept recognition accuracy, while the CNN model only obtains 44.21% accuracy. These results verify the reusability of ASL and demonstrate that the subconcept recognition models are reusable to new tasks that share partial subconcepts with the source task. Indeed, the pre-trained subconcept recognition models can be deemed to learnware [5]. Learnware views a pre-trained model as a product with the specification.

Experiments on multi-instance learning. As discussed in Section 2, ASL can degenerate into multi-instance learning by modifying two places. First, since the instances of an example are presumed to be existed and partitioned well in the first place, the subconcept set selection of ASL is needless for the multi-instance learning. Second, since most multi-instance learning studies assume that if a bag contains a positive instance and then the bag is positive, the logical abduction module's logical rules

Table 3 Classification results on five classical multi-instance learning datasets. Experiments were run five times, and an average of the classification accuracy (± a standard error of a mean, bold text represents the best result) is reported

Method	Musk1	Musk2	Fox	Tiger	Elephant
mi-SVM [36]	0.874±N/A	0.836±N/A	0.582±N/A	0.784±N/A	0.822±N/A
MI-SVM [36]	0.779±N/A	0.843±N/A	0.578±N/A	0.840±N/A	0.843±N/A
MI-Kernel [37]	**0.880**±0.031	**0.893**±0.015	**0.603**±0.028	0.842±0.010	0.843±0.016
EM-DD [38]	0.849±0.044	**0.869**±0.048	**0.609**±0.045	0.730±0.043	0.771±0.043
mi-Graph [21]	**0.889**±0.033	**0.903**±0.039	**0.620**±0.044	**0.860**±0.037	**0.869**±0.035
miVLAD [39]	**0.871**±0.043	**0.872**±0.042	**0.620**±0.044	0.811±0.039	**0.850**±0.036
miFV [39]	**0.909**±0.040	**0.884**±0.042	**0.621**±0.049	0.813±0.037	**0.852**±0.036
Attention [14]	**0.892**±0.040	**0.858**±0.048	**0.615**±0.043	**0.839**±0.022	**0.868**±0.022
ASL	**0.886**±0.024	**0.847**±0.041	0.602±0.031	0.832±0.043	**0.838**±0.036

become simplified. To further verify the effectiveness of ASL, we implemented ASL with five multi-instance learning setting benchmarks. Musk1 and Musk2 are the drug activity prediction datasets [35]. A molecule has the desired drug effect if and only if one or more of its conformations bind to the target binding site [14]. Elephant, Fox, and Tiger are animal classification datasets [36]. Positive bags are images that contain the animal patches of interest, and negative bags are images that contain other animals [36]. In our experiments, we use the same architecture as in the model [14] except for the attention module. We remove the attention module by using simple logical rules to infer the holistic label of the bag. If the max score among the instances is large than 0.5, then the holistic label is positive. Table 3 reports the results that demonstrate that our degenerated method is comparable with existing multi-instance learning methods.

6 Conclusion

In this paper, we introduced subconcepts as a bridge to connect machine learning and logical reasoning toward achieving an interpretable and reusable AI. We proposed the ASL approach that can unify machine learning and logical reasoning in actual application scenarios of AI. ASL is good at subconcept set selection according to domain knowledge and can identify high-level subconcepts in low-level cognitive tasks. Extensive experimental results have verified that ASL obtains high accuracy and can interpret classification results. In-depth analyses also show that ASL has small requirements and low dependence on training data. Interestingly, we can reuse the pre-trained subconcept recognition model in new tasks from different environments. In summary, this study has taken a step forward and paved the way for further studies. In future work, it would be interesting to consider the probabilistic logic by inputting the uncertainty of neural networks into the abductive logical reasoning model.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62176139, 61872225, 61876098) and Major Basic Research Project of Natural Science Foundation of Shandong Province (Grant No. ZR2021ZD15).

References

1 Russell S. Unifying logic and probability. Commun ACM, 2015, 58: 88–97
2 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012. 1097–1105
3 Gunning D, Aha D W. DARPA's explainable artificial intelligence (XAI) program. AI Mag, 2019, 40: 44–58
4 Muggleton S H, Schmid U, Zeller C, et al. Ultra-strong machine learning: comprehensibility of programs learned with ILP. Mach Learn, 2018, 107: 1119–1140
5 Zhou Z H. Learnware: on the future of machine learning. Front Comput Sci, 2016, 10: 589–590
6 Zhou Z-H. Abductive learning: towards bridging machine learning and logical reasoning. Sci China Inf Sci, 2019, 62: 076101
7 Dai W Z, Xu Q, Yu Y, et al. Bridging machine learning and logical reasoning by abductive learning. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 2811–2822
8 Manhaeve R, Dumancic S, Kimmig A, et al. DeepProbLog: neural probabilistic logic programming. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 3749–3759
9 de Raedt L, Kimmig A, Toivonen H. ProbLog: a probabilistic prolog and its application in link discovery. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, Hyderabad, 2007. 2462–2467
10 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436
11 Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 3856–3866
12 Wolpert D H. Stacked generalization. Neural Networks, 1992, 5: 241–259
13 Zhou Z H, Feng J. Deep forest: towards an alternative to deep neural networks. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017. 3553–3559

14 Ilse M, Tomczak J M, Welling M. Attention-based deep multiple instance learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 2132–2141
15 Wang X, Yan Y, Tang P, et al. Revisiting multiple instance neural networks. Pattern Recognit, 2018, 74: 15–24
16 Carbonneau M A, Cheplygina V, Granger E, et al. Multiple instance learning: a survey of problem characteristics and applications. Pattern Recognition, 2018, 77: 329–353
17 Yang S J, Jiang Y, Zhou Z H. Multi-instance multi-label learning with weak label. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, 2013
18 Sun Y Y, Ng M K, Zhou Z H. Multi-instance dimensionality reduction. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010
19 Zhou Z H, Zhang M L, Huang S J, et al. Multi-instance multi-label learning. Artif Intell, 2012, 176: 2291–2320
20 Wang W, Zhou Z H. Learnability of multi-instance multi-label learning. Chin Sci Bull, 2012, 57: 2488–2491
21 Zhou Z H, Sun Y Y, Li Y F. Multi-instance learning by treating instances as non-IID samples. In: Proceedings of the 26th Annual International Conference on Machine Learning, 2009. 1249–1256
22 Mathieu E, Rainforth T, Siddharth N, et al. Disentangling disentanglement in variational autoencoders. 2018. ArXiv:1812.02833
23 Burgess C P, Matthey L, Watters N, et al. MONet: unsupervised scene decomposition and representation. 2019. ArXiv:1901.11390
24 Locatello F, Bauer S, Lucic M, et al. Challenging common assumptions in the unsupervised learning of disentangled representations. 2018. ArXiv:1811.12359
25 Dong H, Mao J, Lin T, et al. Neural logic machines. 2019. ArXiv:1904.11694
26 Shanahan M, Nikiforou K, Creswell A, et al. An explicitly relational neural network architecture. 2019. ArXiv:1905.10307
27 de Raedt L, Kimmig A. Probabilistic (logic) programming concepts. Mach Learn, 2015, 100: 5–47
28 Koller D, Friedman N, Džeroski S, et al. Introduction to Statistical Relational Learning. Cambridge: MIT Press, 2007
29 Kakas A C, Kowalski R A, Toni F. Abductive logic programming. J Logic Computation, 1992, 2: 719–770
30 Yu Y, Qian H, Hu Y Q. Derivative-free optimization via classification. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016
31 Muggleton S H, Lin D, Tamaddoni-Nezhad A. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. Mach Learn, 2015, 100: 49–73
32 Bratko I. Prolog Programming for Artificial Intelligence. Mississauga: Pearson Education Canada, 2012
33 Xian Y, Lampert C H, Schiele B, et al. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 2251–2265
34 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
35 Dietterich T G, Lathrop R H, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artif Intelligence, 1997, 89: 31–71
36 Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. In: Proceedings of Advances in Neural Information Processing Systems, Vancouver, 2002. 561–568
37 Gärtner T, Flach P A, Kowalczyk A, et al. Multi-instance kernels. In: Proceedings of the 19th International Conference on Machine Learning, 2002. 179–186
38 Zhang Q, Goldman S A. EM-DD: an improved multiple-instance learning technique. In: Proceedings of Advances in Neural Information Processing Systems Vancouver, 2001. 1073–1080
39 Wei X S, Wu J, Zhou Z H. Scalable algorithms for multi-instance learning. IEEE Trans Neural Netw Learn Syst, 2017, 28: 975–987