Safe Abductive Learning in the Presence of Inaccurate Rules

Xiao-Wen Yang[1], Jie-Jing Shao[1], Wei-Wei Tu[2], Yu-Feng Li[1*], Wang-Zhou Dai[1], Zhi-Hua Zhou[1]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
[2]4Paradigm Inc., Beijing, China
{yangxw, shaojj, liyf, daiwz, zhouzh}@lamda.nju.edu.cn, tuww.cn@gmail.com

Abstract

Integrating complementary strengths of raw data and logical rules to improve the learning generalization has been recently shown promising and effective, e.g., abductive learning is one generic framework that can learn the perception model from data and reason between rules simultaneously. However, the performance would be seriously decreased when inaccurate logical rules appear, which may be even worse than baselines using only raw data. Efforts on this issue are highly desired while remain to be limited. This paper proposes a simple and effective safe abductive learning method to alleviate the harm caused by inaccurate rules. Unlike the existing methods which directly use all rules without correctness checks, it utilizes them selectively by constructing a graphical model with an adaptive reasoning process to prevent performance hazards. Theoretically, we show that induction and abduction are mutually beneficial, and can be rigorously justified from a classical maximum likelihood estimation perspective. Experiments on diverse tasks show that our method can tolerate at least twice as many inaccurate rules as accurate ones and achieve highly competitive performance while other methods can't. Moreover, the proposal can refine inaccurate rules and works well in extended weakly supervised scenarios.

Introduction

Recently, in order to offer a better understanding of the learning systems and improve the learning generalization, complementary integration of raw data and symbolic rules in a favorable way, has become an active branch and shown to be promising (Raedt et al. 2020; Besold et al. 2021) .

Neural-symbolic learning (NeSy) (Garcez et al. 2019; Sarker et al. 2021; Cunnington et al. 2022) focusing on integrating logical reasoning into the neural networks, has been studied for decades. Deep neural networks (LeCun, Bengio, and Hinton 2015) serve as low-level perception models to translate raw inputs into symbolic concepts of practical meaning, while the knowledge base constrains both the intermediate symbolic concepts and the final target using logical rules. Algorithms include DeepProblog (Manhaeve et al. 2018), NeurASP (Yang, Ishay, and Lee 2020), and LTN (Badreddine et al. 2022), etc. Abductive learning (ABL)

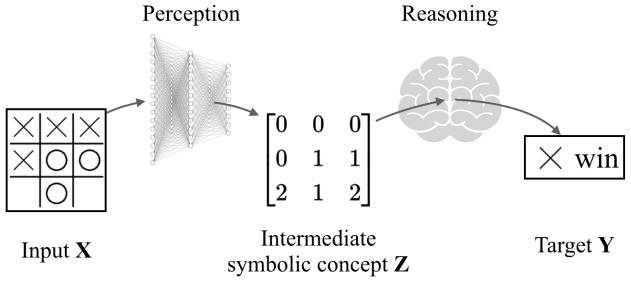

Figure 1: An example of the complementary integration of perception and reasoning for the Tic-Tac-Toe task. The input is a snapshot of an endgame, and the goal is to judge whether "x" wins. Experts provide rules of victory (i.e., when "x" has one of 8 possible ways to create a "three-in-a-row").

(Zhou 2019) is one recent generic and effective framework that bridges any kind of machine learning algorithms and logical reasoning by using inconsistency minimization to construct pseudo-labels of the intermediate symbolic concepts. The coordination of these two modules makes use of the powerful perception ability of learning models and the logical reasoning ability of the knowledge base at the same time, thus enhancing the interpretability and generalization of machine learning models. The description of these ideas using intermediate symbolic concepts as bridges is briefly shown in Figure 1. The performance has been reported that it achieves highly competitive performance to the pure learning models. Algorithms include ABLSim(Huang et al. 2021b), SS-ABL (Huang et al. 2020) and ABL-KG (Huang et al. 2023), etc.

The positive results mentioned above, however, rely on the fundamental assumption that the logical rules are consistently accurate. Such an assumption is difficult to hold in many practical applications because the experts may make mistakes or machine-generated rules may be inaccurate. For example, in the text sentiment analysis task, sentiment lexicons are often built automatically (Lu et al. 2011) which would unavoidably contain inaccurate information. In addition, many works (Chen, Jia, and Xiang 2020; Sadeghian et al. 2019) use deep learning techniques to mine rules from large knowledge graphs, and the uncertainty of neural net-

*Yu-Feng Li is the corresponding author.

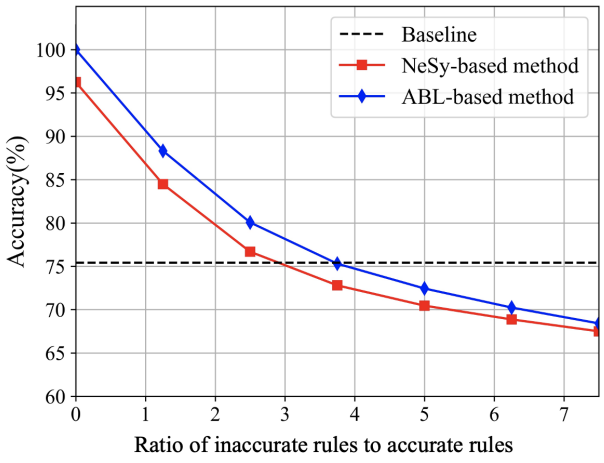

Figure 2: The performance degradation of the NeSy-based method (Badreddine et al. 2022) and the ABL-based method (Huang et al. 2021b) as the ratio of inaccurate rules to accurate rules increases on the Tic-Tac-Toe dataset.

works will introduce inaccurate rules, not to mention the imperfectness of knowledge graphs themselves (Paulheim 2017; Huang et al. 2022, 2023; Liu et al. 2024). Faced with these dilemmas, ABL or NeSy no longer works well and may even be accompanied by severe performance degradation. That is, ABL or NeSy is even worse than a simple end-to-end baseline model using only raw data, as illustrated in Figure 2. Such phenomena undoubtedly go against the expectation and limit their effectiveness in various practical tasks. However, to our best knowledge, the efforts in this aspect remain to be limited.

In this paper, we focus on ABL and try to build a safe ABL algorithm, that is to say, ABL using extra rules (some may be inaccurate) will not be inferior to a simple end-to-end model using raw data only. After the proposal of ABL, the follow-up work mainly focuses on enhancing the quality of pseudo-label, e.g., ABLSim (Huang et al. 2021b) and expanding the capacity of the knowledge base, e.g., GABL (Cai et al. 2021) and there is little research on safe ABL yet. There are also discussions in the field of machine learning about safe weakly supervised learning (Guo et al. 2020; Li, Guo, and Zhou 2021; Zhou, Jing, and Li 2024). Although they adopt the same definition of safeness, they could not be applied to our problem setting because they are purely based on learning models and do not take rules into account. Therefore, to alleviate the performance degradation caused by inaccurate rules, new proposals are desired.

To this end, this paper presents a simple and effective safe ABL framework **Safe-ABL**. Unlike the existing methods (Manhaeve et al. 2018; Badreddine et al. 2022; Yang, Ishay, and Lee 2020; Huang et al. 2021b) which directly use all rules without correctness checks, it uses them selectively to prevent performance hazards. Specifically, we construct a graphical model with an adaptive reasoning process and we argue that the intermediate symbolic concepts can be generated from the abduction process from logical rules and the induction process from raw data. Our method tries to minimize the discrepancy between the generated distribution from both the induction and abduction processes. Theoretically, we show that induction and abduction are mutually beneficial, and can be rigorously justified from a classical maximum likelihood estimation perspective and thus let the perception model and rules rectify each other. Experiments on diverse tasks show that, unlike existing methods that tend to cause severe performance degradation, our new method could tolerate at least twice as many inaccurate rules as accurate ones and achieve highly competitive performance. Moreover, our proposal is able to refine rule quality and can work well in extended weakly supervised scenarios such as the semi-supervised scenario.

Brief Introduction to ABL and NeSy

Abductive learning (Zhou 2019) enables the joint optimization of machine learning and logical reasoning by using inconsistency minimization. It focuses on how to deal with the intermediate symbolic concepts that serve as pseudo-labels for learning and variables for abduction. The pseudo-label can help update the machine learning model, and the abduction searches for the most suitable revised pseudo-label to minimize the inconsistency between raw data and the knowledge base. ABL has many variants. Cai et al. (2021) extends the ABL framework to exploit the logical domain knowledge base represented by groundings. Huang et al. (2021b) uses a similarity-based consistency metric to select the final pseudo-label from all possible abduction results which makes the optimization of the ABL framework faster and more stable. Huang et al. (2020) solved the theft judicial sentencing problem using the ABL framework in a semi-supervised setting. All of these variants assume that the knowledge base is strictly accurate, which is not the case in real applications. To solve the performance degeneration caused by inaccurate rules, new approaches are desired.

Neural-symbolic learning (Besold et al. 2021; Raedt et al. 2020) is concerned with designing algorithms to bridge perception and reasoning. Many Methods (Yang, Lee, and Park 2022; Xu et al. 2018; Fischer et al. 2019; Huang et al. 2021a) treat logical rules as constraints, which serve as effective regularization for training networks. Xu et al. (2018) designs a loss function forcing the output of networks to conform more closely to logic constraints. Other methods (Yang, Lee, and Park 2022; Fischer et al. 2019) follow similar lines of thought to deal with different types of logic constraints. These methods weaken the role of logical systems and thus may lose the strong power of logical reasoning. Moreover, many methods (Badreddine et al. 2022; Manhaeve et al. 2018) have focused on how neural networks can blend into existing tools for logical reasoning. Badreddine et al. (2022) extends fuzzy logic with neural predicate into a fully differentiable logical language called Real Logic. DeepProbLog (Manhaeve et al. 2018) is a probabilistic logic programming language that similarly incorporates deep learning using neural predicates. The goal of these methods is to build complete logical systems using a differentiable way. However, they all ignore the impact of rules for machine learning leading to the collapse of model learning when faced with

inaccurate rules.

There are two other typical paradigms for integrating machine learning and logical reasoning. The Probabilistic Logic Program (PLP) aims to expand FOL to accommodate probabilistic groundings, allowing for probabilistic inference. Statistical Relational Learning (SRL) aims to create a probabilistic graphical model based on domain knowledge expressed in FOL, sharing a similar motivation to NeSy that external domain knowledge is utilized to establish an interpretable neural structure. Different from all these paradigms, ABL can jointly optimize machine learning and logical reasoning through inconsistency minimization.

The Proposed Safe-ABL Framework

To alleviate the performance degradation caused by inaccurate rules, we propose a simple and effective ABL framework Safe-ABL. Unlike the existing methods which directly use all rules, we give them weights to represent the probabilities that each rule participates in abduction. Our main idea is that errors in the perception model and rules can be rectified mutually. We implement this idea by constructing a graphical model with an adaptive reasoning process of intermediate symbolic concepts and then minimize the discrepancy between the generated distribution from the induction and abduction processes. Theoretical results show that our optimization can be rigorously justified from a classical maximum likelihood estimation perspective. Figure 3 illustrates the Safe-ABL framework. In this section, we first give the Safe-ABL framework with a graphical model and its analysis, and then the running time complexity of our proposal is accelerated for more practical applications.

Notations and Problem Setting

Considering standard supervised learning, we define an *input space* $\mathcal{X}$ and an *output space* $\mathcal{Y}$ of size K. A distribution $\mathcal{D}$ is defined on space $\mathcal{X} \times \mathcal{Y}$. We sample a training dataset $D = \{(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$ from this distribution. In addition, we are given a discrete *symbol space* $\mathcal{Z}$. The symbol space can be seen as cartesian product of a series of feature space $\mathcal{Z} = \prod_{i=1}^{s} \mathcal{Z}_i$. Intermediate symbol $\mathbf{z} = (z^1, z^2, ..., z^s)$ from $\mathcal{Z}$ has exact meaning which human can understand. For example, the corresponding symbol of one Tic-Tac-Toe endgame picture is a 3×3 matrix (i.e., $s = 9$), the element of which represents 'x', 'o', or blank. A knowledge base KB which represents a set of logical rules is provided by experts. As discussed previously, rules from this knowledge base may be inaccurate. A filter function $g : KB \to \{0, 1\}$ which can judge whether the rule is accurate, is needed. Our goal is to learn a model $h : \mathcal{X} \to \mathcal{Y}$. Thanks to the knowledge base KB provided by experts, we can first learn a perception model $f : \mathcal{X} \to \mathcal{Z}$, then the intermediate symbol $\mathbf{z}$ is fed into KB to reason to the final target $\mathbf{y}$. Moreover, to refine the knowledge base for its better use, we should also learn the filter function g.

To better understand our method, we first introduce the principle of inductive reasoning and abductive reasoning. Inductive reasoning or induction is the primary task of machine learning and can infer general knowledge from specific raw data. In our paper, perception is a way of induction. Abduction or abductive reasoning (Josephson and Josephson 1996; Shelley 2012) is a basic form of logical inference different from induction and deduction. Considering a real scene. One morning, you go outside and the ground is wet, and you guess it must have rained last night. The way of your think is abduction which is to identify causes that are the most promising explanations for current observations. We call all the possible explanations as abduction results $\mathbb{A}$. Formally, $\mathbb{A} := \{\mathbf{z} \in \mathcal{Z} : \mathbf{z} \cup KB \vDash \mathbf{y}\}$, symbol $\vDash$ means logical implication. For example in Tic-Tac-Toe game, If we observe that "x" wins, and we know the winning rules, we can abduce all possible endgame states.

Framework Formulation with Graphical Model

In order to model the impact made by inaccurate rules, we construct a graphical model to describe the reasoning process of intermediate symbolic concepts which is an effective way. Figure 4 demonstrates our graphical model. We argue that each rule in the knowledge base guides the abduction of intermediate symbolic concepts together with the label. Each directed edge pointed from the knowledge base represents the direction of the abduction. In addition, a random factor ϵ is introduced which is to randomly select a certain $\mathbf{z}$ from abduction results $\mathbb{A}$. To be specific, we can give meaning to ϵ such as the random index of the list of abduction results. Then given a specific instance $\mathbf{x}$, the perception model will induce the intermediate symbolic concept $\mathbf{z}$. The directed edge pointed from $\mathbf{x}$ represents the direction of the induction. Take the example of a graphical model in the Tic-Tac-Toe game, when we have the winning rules (i.e., KB) and know 'x' wins (i.e., $\mathbf{y}$), we can abduce all possible endgame states (i.e., $\mathbb{A}$). The random factor ϵ will decide which endgame state will be finally chosen. When given the snapshot of this board (i.e., $\mathbf{x}$), we can induce the endgame state (i.e., $\mathbf{z}$) by the perception model.

However, having all rules point to $\mathbf{z}$ is dangerous because rules may be inaccurate. To conduct safe abduction, our key idea is to extend the graphical model by giving weights for the edges between rules and $\mathbf{z}$. Firstly, we classify rules into two parts: expert-convinced rules KB_t and rules that may be inaccurate KB_w, where the latter ones require careful exploitation and refinement. Obviously, we have $KB_t \cap KB_w = \emptyset$ and $KB_t \cup KB_w = KB$. Then, for rules in KB_w, we set weights for its edge pointing to $\mathbf{z}$. Formally, we set a parameter $\omega_i \in [0, 1]$ for each rule $r_i \in KB_w$. We write all the ω_i in the unitive form ω.

These parameters represent the probabilities that each rule participates in abduction if it does not contradict $\mathbf{z}$. Here we assume that these probabilities are invariant no matter what $\mathbf{z}$ is. Generally speaking, we want accurate rules to have a higher probability of participating in abduction, and inaccurate rules to have a lower probability. Such an observation is realistic and help refine the rule quality. Before training, since we don't know the correctness of each rule, we set $w_i^{(0)} = 0.5, i = 1, 2..., |KB_w|$. The superscript represents the training rounds. According to the graphical model, we can define the abduction probability $Q_\omega(\mathbf{z}|\mathbf{y}, KB)$ of each intermediate symbolic concept $\mathbf{z}$. This abduction probability

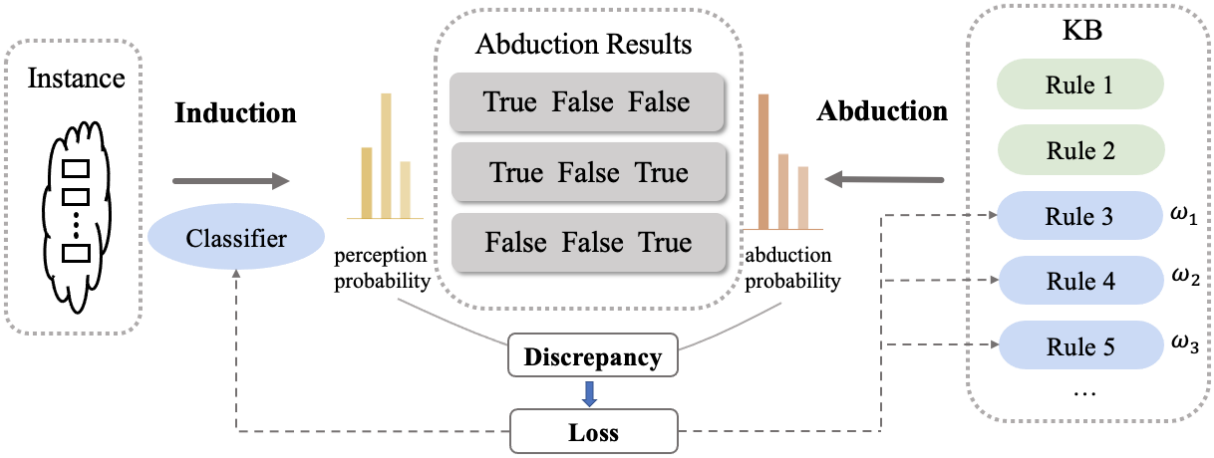

Figure 3: Illustration of the Safe-ABL framework. Green rules are expert-convinced rules, and the blue ones may be inaccurate. The dashed arrow represents the path of the backpropagation update.

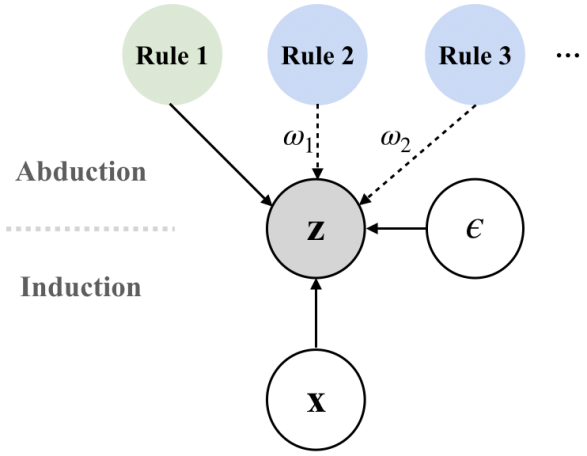

Figure 4: Graphical model of the reasoning process. Dashed edges mean the abduction process may be inaccurate. We set parameters to check them, indicating the probability of their accuracy. The green node means expert-convinced rules.

is the conditional probability of $\mathbf{z}$ given target $\mathbf{y}$ and knowledge base KB. We denote $Q_\omega(\mathbf{z}|\mathbf{y}) := Q_\omega(\mathbf{z}|\mathbf{y}, KB)$.

Here we assume the random factor is sampled from a uniform distribution. This is reasonable due to the lack of prior knowledge of the task structure. So we can calculate the abduction probability $Q_\omega(\mathbf{z}|\mathbf{y})$ for each $\mathbf{z}$. This probability is equal to at least one rule that does not contradict $\mathbf{z}$ participating in abduction. Mathematically speaking,

$$Q_\omega(\mathbf{z}|\mathbf{y}) \propto \begin{cases} 1 - \prod_{r_i \in KB_w} \left(1 - \mathbb{I}\left(r_i \cup \mathbf{z} \cup \mathbf{y} \nvDash \bot\right) w_i\right) & \mathbf{z} \in \mathbb{A}_t, \\ 0 & \mathbf{z} \notin \mathbb{A}_t. \end{cases} \tag{1}$$

Here, $\mathbb{I}$ is the indicator function and $\mathbb{A}_t := \{\mathbf{z} \in \mathcal{Z} : \mathbf{z} \cup KB_t \vDash \mathbf{y}\}$. We can calculate the final value of this probability by normalization.

Optimization for Safe Abduction

In this section, we formalize our target to not only help to learn the perception model but to refine the rule quality.

Considering the edge from $\mathbf{x}$ to $\mathbf{z}$ is an induction process in the graphical model, we can use a perception model to estimate the probability of each $\mathbf{z}$. We use a neural network that can map input $\mathbf{x}$ into a distribution $P_\theta(\mathbf{z}|\mathbf{x})$. Now we have two distributions depicting intermediate symbolic concepts from perception and reasoning modules. Both distributions are characterizations of the intermediate symbolic concepts which should be as consistent as possible intuitively, so we need to minimize their discrepancy. Based on this idea, we design a loss function L and our target is to minimize it, which can be formalized as follows:

$$\begin{aligned} \theta^*, \omega^* &= \arg\min_{\theta,\omega} L(\theta, \omega) \\ &= \arg\min_{\theta,\omega} -\frac{1}{n} \sum_{i=1}^{n} \log\Big(\sum_{z \in \mathbb{A}_t} P_\theta(\mathbf{z}|\mathbf{x}_i) Q_\omega(\mathbf{z}|\mathbf{y}_i)\Big) \\ \text{s.t.} \quad & 0 \le \omega_j \le 1, j = 1, 2, ..., |KB_w| \end{aligned} \tag{2}$$

Intuitively, if we have a well-initialized perception model, pulling in these two distributions helps to distinguish the inaccurate rules in the knowledge base, and in turn, when the knowledge base is good enough, it can promote the convergence of the perception model to the global optimal. Since all operations are differentiable, we can solve it by gradient descent in each training step.

$$\begin{aligned} \theta^{(k+1)} &= \theta^{(k)} - \eta_1 \cdot \nabla_\theta L \\ \omega^{(k+1)} &= \text{Clamp}\left(\omega^{(k)} - \eta_2 \cdot \nabla_\omega L, 0, 1\right) \end{aligned} \tag{3}$$

$\text{Clamp}(\cdot, 0, 1)$ limits the value of ω between $[0, 1]$. η_1 and η_2 are the learning rates.

We further analyze our optimization objective from the theoretical perspective.

Theorem 0.1. *Assume that $\mathbb{P}(\mathbf{y}|KB,\omega)$ and $\mathbb{P}(\mathbf{z}|KB,\omega)$ are both uniform distributions. Our optimization objective (2) is equivalent to maximum likelihood estimation, i.e.,*

$$\theta^*,\omega^* = \arg\max_{\theta,\omega} \prod_{i=1}^{n} \mathbb{P}(\mathbf{y}_i|\mathbf{x}_i, KB, \theta, \omega) \tag{4}$$

The assumption of this theorem is reasonable because the prior distributions of $\mathbf{y}$ and $\mathbf{z}$ are independent of the knowledge base and the parameters in the knowledge base. This theorem provides a clear statistical guarantee for our optimization objective. It suggests that by leveraging the raw data, the disambiguation process between the perception model and inaccurate rules can be facilitated, leading to their mutual promotion. The proof is provided in Appendix.

After training, we get the convergence parameters for each rule in KB_w. We set a threshold $\delta \in [0,1]$ to distinguish which rule is accurate. Formally, if $\omega_i \geq \delta$, we treat the i'th rule as the accurate one while $\omega_i < \delta$ opposite. This means the filter function $g(r_i) = \mathbb{I}(w_i \geq \delta)$. In order to be consistent with the initial parameters of the training, we fix $\delta = 0.5$. In some other cases, rules may inherently be fuzzy, and our algorithm can quantify this fuzziness. Therefore, we have attained our aim of refining the quality of the rules.

Running Time Acceleration

In this section, we try to make the abduction results $\mathbb{A}_t$ smaller based on the confidence of the neural network output, thereby reducing the training cost. Since we should compute $Q(\mathbf{z}|\mathbf{x})$ for all the possible $\mathbf{z}$, it is intractable if the cardinality of $\mathbb{A}_t$ is large enough. The confidence of the neural network output helps to make $\mathbb{A}_t$ smaller.

For each symbol space $\mathcal{Z}_k$, the neural network can output the probability of each symbol representing model's confidence. So, it is a good way to rule out symbols with low confidence. Formally, we select a subset $\tilde{\mathbb{A}}_t$ of $\mathbb{A}_t$.

$$\tilde{\mathbb{A}}_t = \{\mathbf{z} = (z^1, z^2, ..., z^s) | z \in \mathbb{A}_t \wedge NN(z^i) > \sigma\} \tag{5}$$

$NN(z^i)$ means the predicted probability of the neural network for the feature z^i. The σ is the hyperparameter that selects with high confidence and we set $\sigma = 0.99$ in all our experiments. Since we reduce the size of $\mathbb{A}_t$, we need to normalize $Q_\omega(\mathbf{z}|\mathbf{x})$ again. We denote the new value as $\tilde{Q}_\omega(\mathbf{z}|\mathbf{x})$. So our goal becomes

$$\begin{aligned} \theta^*,\omega^* &= \arg\min_{\theta,\omega} L(\theta,\omega) \\ &= \arg\min_{\theta,\omega} -\frac{1}{n}\sum_{i=1}^{n} \log(\sum_{\mathbf{z}\in\tilde{\mathbb{A}}_t} P_\theta(\mathbf{z}|\mathbf{x}_i)\tilde{Q}_\omega(\mathbf{z}|\mathbf{y}_i)) \\ \text{s.t.} \quad & 0 \leq \omega_j \leq 1, j = 1, 2, ..., |KB_w| \end{aligned} \tag{6}$$

Experiments

In this section, we present experimental comparison results of our proposed method on different tasks, including MNIST addition (Manhaeve et al. 2018), Tic-Tac-Toe game, and legal dispute focus identification. The purpose of the experiments is to answer the following four questions.

Q1: Whether our method is safe with inaccurate rules?
Q2: Is our method able to improve the quality of rules?
Q3: Can our acceleration technique successfully work?
Q4: Can our method work well in extended weakly supervised scenarios?

The first two tasks are used to answer the first three questions. The legal dispute focus identification task is to answer the fourth question, which contains real data scenarios such as multi-label and semi-supervised data. To ensure the fairness of our experiments, all methods share the same knowledge base and a pre-trained perception model. All experiments are repeated five times with Nvidia Tesla V100 GPU.

MNIST Addition

This task was first introduced by DeepProblog (Manhaeve et al. 2018) which contains two subtasks, **Single-digit** and **Multi-digit**. The input of the first subtask is a pair of MNIST images (LeCun et al. 1998), and the output is the sum of the individual digits. The input of the second subtask is a list of four MNIST images that make two tens digits in pairs and the output is their sum.

The original rule in the knowledge base is the addition operation. In our environment, the knowledge base contained inaccurate rules. For the single-digit subtask, the knowledge base contains four rules, namely addition, subtraction, multiplication, and division. For more complex multi-digit subtasks, the knowledge base contains two rules: addition and multiplication. These rules partially match the final result (for example, $2+2=4$ and $2\times 2=4$), but they do not fully match the requirement of the task and are therefore inaccurate, making this task difficult to solve.

We compare our approach with the end-to-end baseline, existing state-of-the-art neural-symbolic approaches, and abductive learning approaches. NeSy methods include Logic Tensor Network (LTN) (Badreddine et al. 2022), DeepProblog (Manhaeve et al. 2018). ABL methods include the original abductive learning(ABL) (Dai et al. 2019) and ABL with similarity (ABLsim) (Huang et al. 2021b), a faster and more effective variant of ABL. We choose LeNet-5 (LeCun et al. 1998) as the perception model for all the methods and we initialize it with a subset of labeled samples. For these two subtasks, the initialized perception model performance is 10.4% and 20.3%, respectively. The reason we adopt the different initial models is due to the different difficulty levels of these two tasks.

To answer Q1, we demonstrate comparative performance via end-to-end classification accuracy. Due to the inaccurate rules provided by the knowledge base, none of the existing methods can be tested except the baseline method. For fair comparison, we assume that all existing methods have the accurate knowledge base during the testing phase. This compromise can also verify the superiority of our method for improving the knowledge base. As shown in Table 1, the performance of the current method is much lower than the baseline due to inaccurate rules. The main reason is that the predictions of existing methods all heavily depend on the

Method	Single-digit	Multi-digit
Baseline	96.78 ± 0.21	60.46 ± 2.67
LTN	79.01 ± 0.41	53.43 ± 2.62
DeepProbLog	21.81 ± 7.12	48.27 ± 0.52
ABL	20.82 ± 0.94	16.66 ± 0.85
ABLSim	86.40 ± 9.72	57.09 ± 0.55
Safe-ABL(Ours)	**99.04 ± 0.11**	**98.06 ± 0.14**

Table 1: Classification accuracies (%mean ± std) for MNIST Addition task of different methods.

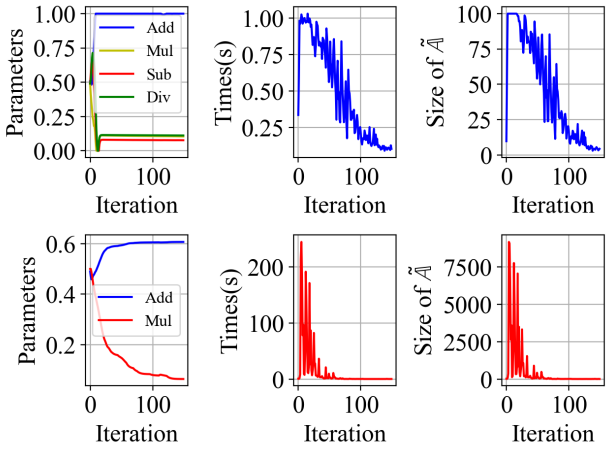

Figure 5: Curve results for MNIST Addition. The top three are for Single-digit and the bottom three are for Multi-digit.

correctness of the rules and rules for this task are inherently conflicting which can easily mislead the perception model. In contrast, our method guarantees high performance even if the knowledge base contains inaccurate rules. Therefore, our method behaves much more robustly to inaccurate rules which means our method is safer.

To answer Q2, we demonstrate the variation curve of the weight parameter of the rules in Figure 5. It can be observed that with the deepening of training, the weight of accurate rules gradually increases, while the weight of inaccurate rules gradually decreases. Therefore, we can successfully locate the inaccurate rules and keep the accurate ones, and finally achieve the purpose of refining the rule quality.

To answer Q3, we record the average size of the accelerated abduction results $\tilde{\mathbb{A}}$ for each iteration and the time taken for each training iteration. As shown in Figure 5, the average size of $\tilde{\mathbb{A}}$ is gradually decreasing due to our acceleration technique, resulting in a decrease in the time spent on each training iteration. Therefore, our self-acceleration technique can alleviate the problem of high abduction consumption while maintaining high performance.

Tic-Tac-Toe Game

The Tic-Tac-Toe game is a famous game around the world for two players who take turns marking the spaces in a 3×3 grid with 'x' or 'o'. The Tic-Tac-Toe game is first introduced

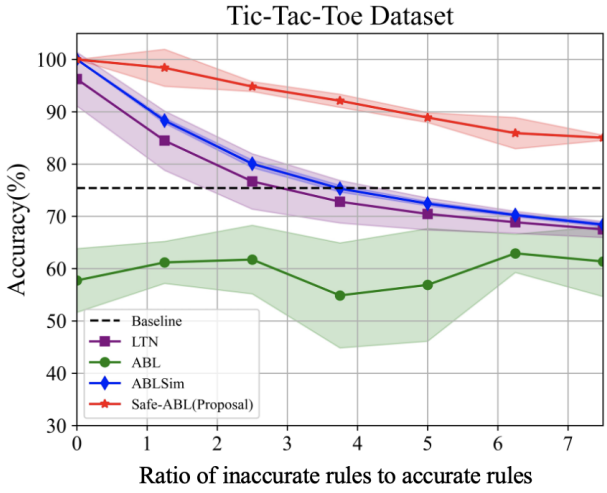

Figure 6: Classification accuracy(%) of different methods on the Tic-Tac-Toe varying the number of inaccurate rules. Shaded regions indicate standard deviation.

by UCI and Kaggle [1]. The target of this task is to judge whether 'x' wins given the endgame state. The original task is easy and no knowledge base is introduced. Here we transform it into an image binary classification task. We use a pre-trained LeNet-5 (LeCun et al. 1998) as the perception model for all the methods.

Then we introduce the knowledge base. The expert-convinced rules in the knowledge base are that 'x' play first and 'x' takes at least three steps. Each uncertain rule is a triplet of positions representing victory. For example, we number the 3×3 grids in row order and an uncertain rule triplet [0,4,8] (i.e., main diagonal) means if 'x' exists in these three positions simultaneously then 'x' wins. Out of the rules list, only 8 rules can be strictly accurate, while many other rules are found to be inaccurate because a significant number of samples are unable to obey them.

To answer Q1, we adjust the number of inaccurate rules from 0 to 60 to compare the performance degradation of different methods. Inaccurate rules are randomly generated and mixed in the uncertain rules list. Due to the increasing number of rules, some algorithms with high costs such as DeepProblog (Manhaeve et al. 2018) cannot be applied to this task, so LTN (Badreddine et al. 2022) is choosen as the typical NeSy method. Results are shown in Figure 6. When varying the number of inaccurate rules, the performance of LTN and ABLSim (Huang et al. 2021b) decrease significantly and below the baseline. Nevertheless, our method maintains high accuracy all the time. This phenomenon indicates that our method is much safer than other methods.

To answer Q2, we view the refinement of the knowledge base as a binary classification task that judges the inaccurate rules. Due to the imbalance in the number of accurate and inaccurate rules, we employ a threshold-free metric: the

[1]https://www.kaggle.com/datasets/aungpyaeap/tictactoe-endgame-dataset-uci

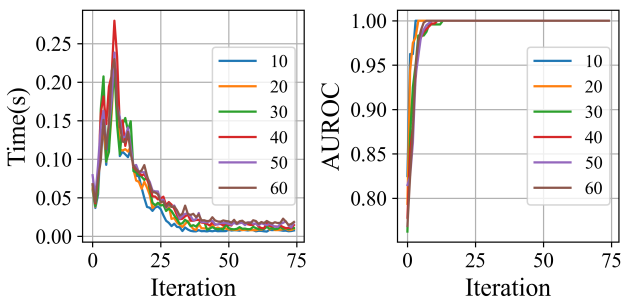

Figure 7: AUROC curve and Time curve of the Tic-Tac-Toe. The legend represents the number of inaccurate rules.

area under the receiver operating characteristic curve (AUROC). On the right side of the Figure 7, the AUROC curve shows the number of erroneous rules that were successfully located. The results show that our method is able to distinguish between accurate and inaccurate rules, and is stable to different numbers of inaccurate rules.

To answer Q3, we show the time spent training for each iteration on the left side of Figure 7. We observe that the training time starts out small, quickly increases, and eventually decreases. The reason why the time cost is small at first is because we take the initial trained model and there will be more high-confidence samples. However, these high-confidence samples may be misclassified and the model will try to accurate them, thus choosing a larger $\tilde{\mathbb{A}}$ during training. Finally, as the model becomes more confident, a smaller $\tilde{\mathbb{A}}$ will be chosen, so training becomes faster and faster.

Legal Dispute Focus Identification

In this section, we introduce a complex symbolic reasoning task: Dispute Focus Identification in the legal domain. The dispute focus, which is crucial to legal judgment and dispute resolution, delineates the primary bone of contention between the plaintiff and defendant. The task takes a Chinese pleading sentence as its input and outputs the type of dispute focus. For instance, divorce litigation may involve a contention over child custody, thus the pleading sentences will include the child support dispute focus. This task comprises 1000 labeled samples and 2000 unlabeled samples, constituting as a semi-supervised learning task. Each pleading sentence may have multiple dispute focuses. Therefore, the task is a multi-label task. For example, a divorce pleading sentence could contend child support and alimony simultaneously. We have followed privacy and data protection consent requirements during the data collection process, employing anonymization techniques for personal names.

The field of jurisprudence offers a vast repertoire of statutory rules that are sourced from the letter of the law. We gather 166 rules to constitute our knowledge base in this task. Due to the lack of expertise in this domain, we cannot ensure the accuracy of these rules. Moreover, many of them are inherently ambiguous. Here is an example of a logical rule in the knowledge base: alimony_dispute(x) $\leftrightarrow$ adopter_dispute(x). While this rule holds valid in most cases, there are always exceptional circumstances that do not conform to the precept, rendering it inaccurate. Therefore, methods that can cope with inaccurate rules are urgently needed.

Method	Micro-F1	Macro-F1
Baseline	69.66 $\pm$ 0.83	64.44 $\pm$ 1.60
Pseudo-Label	70.17 $\pm$ 0.80	63.58 $\pm$ 1.39
Tri-training	67.84 $\pm$ 1.26	58.01 $\pm$ 1.90
SS-ABL	70.48 $\pm$ 0.71	64.18 $\pm$ 0.65
Safe-ABL(Ours)	**70.72** $\pm$ **0.42**	**66.14** $\pm$ **1.32**

Table 2: Micro-F1 score (%) and Macro-F1 score (%) on the Dispute Focus Identification task.

Existing NeSy methods have little extension to weakly supervised scenarios and thus are not directly applicable to this task. SS-ABL(Huang et al. 2020) provides a unified ABL framework for semi-supervised scenarios and thus can be applied to new scenarios as our comparison method. Safe-ABL is compared with baseline, SS-ABL, and two semi-supervised methods, namely, pseudo-labeling methods (Lee et al. 2013) and Tri-training method (Zhou and Li 2005). Baseline methods use only labeled data. All methods use the same pre-trained BERT model (Devlin et al. 2019) as the backbone architecture. We use the Micro-F1 score and the Macro-F1 score as our performance metrics. As shown in Table 2, our method achieves the best performance compared to other methods. More importantly, unlike other semi-supervised methods, which lead to performance degradation, our method does not suffer from this. SS-ABL achieves better performance than the baseline but suffers from inaccurate rules and underperforms our new method. The above results show that when the knowledge base is imprecise or has inaccurate rules, our method is effective and has a promising prospect for realistic tasks.

Conclusion

In this paper, we investigate an important issue in the complementary integration of raw data and logical rules, namely that the performance of existing ABL and NeSy algorithms suffers severely when inaccurate rules exist in the knowledge base. To this end, we propose a new framework safe ABL based on ABL. The effectiveness of the new algorithm has been supported and verified both theoretically and empirically. In theory, we show that our optimization proves to be equivalent to classical maximum likelihood estimation, thus allowing perception models and rules to rectify each other. Empirical studies show that, unlike the existing ABL and NeSy methods, which suffer performance degradation, our method does not encounter this situation and is obviously more robust. Furthermore, our method works well in extended weakly supervised scenarios.

There may be several possible studies worth exploring in the future. For example, it is worth extending larger and more complex knowledge bases which inevitably contain inaccurate rules. Moreover, we will consider the use of rules when dealing with large language models in the future.

Acknowledgments

This research was supported by the National Key R&D Program of China (2022YFC3340901) and the National Natural Science Foundation of China (62176118, 62206124).

References

Badreddine, S.; d'Avila Garcez, A. S.; Serafini, L.; and Spranger, M. 2022. Logic Tensor Networks. *Artificial intelligence*, 303: 103649.

Besold, T. R.; d'Avila Garcez, A. S.; Bader, S.; Bowman, H.; Domingos, P.; Hitzler, P.; Kühnberger, K.; Lamb, L. C.; Lima, P. M. V.; de Penning, L.; Pinkas, G.; Poon, H.; and Zaverucha, G. 2021. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 1–51.

Cai, L.; Dai, W.; Huang, Y.; Li, Y.-F.; Muggleton, S. H.; and Jiang, Y. 2021. Abductive Learning with Ground Knowledge Base. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 1815–1821.

Chen, X.; Jia, S.; and Xiang, Y. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141.

Cunnington, D.; Law, M.; Lobo, J.; and Russo, A. 2022. Inductive Learning of Complex Knowledge from Raw Data. *arXiv preprint arXiv:2205.12735*.

Dai, W.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging Machine Learning and Logical Reasoning by Abductive Learning. In *Advances in Neural Information Processing Systems*, 2811–2822.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Fischer, M.; Balunovic, M.; Drachsler-Cohen, D.; Gehr, T.; Zhang, C.; and Vechev, M. T. 2019. DL2: Training and Querying Neural Networks with Logic. In *Proceedings of the 36th International Conference on Machine Learning*, 1931–1941.

Garcez, A. d.; Gori, M.; Lamb, L. C.; Serafini, L.; Spranger, M.; and Tran, S. N. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Guo, L.-Z.; Zhang, Z.-Y.; Y., J.; Li, Y.-F.; and Zhou, Z.-H. 2020. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In *Proceedings of the 37th International Conference on Machine Learning*, 3897–3906.

Huang, J.; Li, Z.; Chen, B.; Samel, K.; Naik, M.; Song, L.; and Si, X. 2021a. Scallop: From Probabilistic Deductive Databases to Scalable Differentiable Reasoning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, 25134–25145.

Huang, J.; Zhao, Y.; Hu, W.; Ning, Z.; Chen, Q.; Qiu, X.; Huo, C.; and Ren, W. 2022. Trustworthy Knowledge Graph Completion Based on Multi-sourced Noisy Data. In *Proceedings of the ACM Web Conference*, 956–965.

Huang, Y.; Dai, W.; Cai, L.; Muggleton, S. H.; and Jiang, Y. 2021b. Fast Abductive Learning by Similarity-based Consistency Optimization. In *Advances in Neural Information Processing Systems*, 26574–26584.

Huang, Y.; Dai, W.; Yang, J.; Cai, L.; Cheng, S.; Huang, R.; Li, Y.-F.; and Zhou, Z.-H. 2020. Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing. In *20th IEEE International Conference on Data Mining*, 1070–1075.

Huang, Y.-X.; Sun, Z.; Li, G.; Tian, X.; Dai, W.-Z.; Hu, W.; Jiang, Y.; and Zhou, Z.-H. 2023. Enabling Abductive Learning to Exploit Knowledge Graph. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI'23)*, 3839–3847.

Josephson, J. R.; and Josephson, S. G. 1996. *Abductive inference: Computation, philosophy, technology*.

LeCun, Y.; Bengio, Y.; and Hinton, G. E. 2015. Deep learning. *Nature*, 521(7553): 436–444.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 896.

Li, Y.-F.; Guo, L.-Z.; and Zhou, Z.-H. 2021. Towards Safe Weakly Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 334–346.

Liu, J.-Q.; Yu, Z.-W.; G, B.; C., D.; F, L.-Y.; Wang, X.-B.; and Zhou, C.-H. 2024. EvolveKG: a general framework to learn evolving knowledge graphs. *Frontiers of Computer Science*, 18(3): 183309.

Lu, Y.; Castellanos, M.; Dayal, U.; and Zhai, C. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, 347–356.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and Raedt, L. D. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems*, 3753–3763.

Paulheim, H. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3): 489–508.

Raedt, L. D.; Dumancic, S.; Manhaeve, R.; and Marra, G. 2020. From Statistical Relational to Neuro-Symbolic Artificial Intelligence. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4943–4950.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In *Advances in Neural Information Processing Systems*, 15321–15331.

Sarker, M. K.; Zhou, L.; Eberhart, A.; and Hitzler, P. 2021. Neuro-symbolic artificial intelligence: Current trends. *arXiv preprint arXiv:2105.05330*.

Shelley, C. 2012. Lorenzo Magnani: Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning. *Minds and Machines*, 22(3): 263–269.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and den Broeck, G. V. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, 5498–5507.

Yang, Z.; Ishay, A.; and Lee, J. 2020. NeurASP: Embracing Neural Networks into Answer Set Programming. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 1755–1762.

Yang, Z.; Lee, J.; and Park, C. 2022. Injecting Logical Constraints into Neural Networks via Straight-Through Estimators. In *International Conference on Machine Learning*, 25096–25122.

Zhou, Z.; Jing, Y.-X.; and Li, Y.-F. 2024. Rts: learning robustly from time series data with noisy label. *Frontiers of Computer Science*, 18(6): 186332.

Zhou, Z.-H. 2019. Abductive learning: towards bridging machine learning and logical reasoning. *Sci. China Inf. Sci.*, 62(7): 76101:1–76101:3.

Zhou, Z.-H.; and Li, M. 2005. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11): 1529–1541.