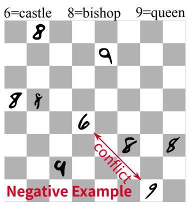
We want to thank the reviewers for the supportive and insightful comments, and we will improve the paper accordingly. In the following we focus on the major technical questions because of the page limit.

To Reviewer #1:

Q1 The technique is very general ... the one problem is the experiment domain, which feels simple and "toy", and should be ideally supplemented with either another simple domain or a more "real" problem. *(Also asked by Rivewer 3)*



ABL-ALP	ABL-CLP-FD
0.976 ± 0.008	0.981 ± 0.006
ABL-CHR	**CNN**
0.979 ± 0.007	0.7355 ± 0.016
Bi-LSTM	
0.513 ± 0.013	

Figure 1: Extended n-queens and the results.

A1 We agree that the proposed framework is general and applicable to many tasks. Therefore, we have made a quick experiment — the *extended n-queens task*, whose inputs are images of chess boards with blanks, queens, castles and bishops (represented by randomly sampled MNIST images), and the labels are whether the state of board is valid. Furthermore, we implemented logical abduction with plain ALP and two popular *constraint logic programming* systems (CHR & CLP-FD). An example and the results are shown in fig. 1.

Q2 What if you were to replace the logical theory learner with something more like metagol?

A2 Thank you for the excellent suggestion. Metagol is implemented by second-order abduction, which fits in our proposed framework very well; it will also enable inducing recursive first-order logical rules and predicate invention. There are many other options, such as constraint logic programming, answer set programming, and so on, as long as they can make logical abduction. Meanwhile, this paper focuses on presenting and verifying the idea of combining perception with reasoning by abduction and trial-and-error search. Thus we will discuss the above as the future work.

Q3 Why do you need the MLP? ... Why can't you just use the logical theory and $p()$ at the final iteration?

A3 We add MLP because the logical theories abduced from *subsamples* of the data might be inaccurate; and the labels may also contain noise. MLP is convenient for dealing with these noise. We will revise to discuss this point.

Q4 Why not have a soft constraint (in eq. 5)? What do you set M to? Is it ad-hoc? *(Also asked by Rivewer 3)*

A4 In the experiments, we set M equals to 10. M defines the step-wise search space on the scale of the abduction and is sufficient to be set a small number. We will revise to discuss this.

Q5 Can you say something about RACOS?

A5 It is a derivative-free optimisation approach like evolutionary algorithm with theoretical ground.

Q6 Why is RBA so much harder than DBA? Is it only because of the perceptual front-end?

A6 It is because of the perceptual front-end, the datasets share the same equation structures. RBA images are more difficult for the CNN network to distinguish, because it doesn't have a large amount of training data.

To Reviewer #2:

Q1 The difference between the proposed approach to DeepProbLog is not entirely clear.

A1 DeepProbLog relies on Algebraic Prolog that has particularly designed operators to *support gradient descent* for jointly optimising the parameters of ProbLog programs and the neural nets; While ABL utilises logical abduction and trial-and-error search to bridge the neural nets with the original Prolog system, *without* using gradient. As the result, our framework inherits the full power of first-order logical reasoning, e.g., it can abduce new logical theories that are not in the background knowledge. Consequently, many existing symbolic AI techniques, such as constraint logic programming, can be directly incorporated in this framework *without* introducing gradients—usually by relaxing logic inference to continuous functions. Thank you for pointing out the issue, we'll add more discussion in the revision.

Q2 ... many interesting details seem to be in the supplementary material. *(Also asked by Rivewer 1)*

A2 Thank you and Reviewer 1 for the nice suggestion, we will re-arrange the paper structure by moving the motivating example and the details about the background knowledge to the main text.

Q3 The comparison with the results achieved by humans is to briefly explained.

A3 We will revise to explain the results. Roughly speaking, it shows that human do not suffer from distinguishing different symbols, while machines are better in checking the consistency of logical theories — in which people are prone to make mistakes. Therefore, machine learning systems should make use of their advantage in logical reasoning.

To Reviewer #3:

Q1 ... the abduced knowledge Δ_C is described as "a set of first-order logical clauses" when eventually it seems to be clear that these could only be ground literals. *(Also asked by Rivewer 1)*

A1 In the equation decipherment tasks, abducing groundings is sufficient for learning the binary operations, so we implemented the abduction with first-order logic. However, our framework does not restrict the order of background knowledge because it utilises the original Prolog system. By exploiting 2nd-order abduction like metagol (as suggested by Reviewer 1), it could learn more complicated theories. We will revise to discuss the above and make it clear.

Q2 The fixed knowledge B cannot be modified by learning. It would be good to show the actual clauses in B.

A2 The programs in the supplementary material are the actual clauses in our codes. They only describe the general structure of the domain, while the more concrete logical theories are learned. How to modify the general background knowledge, i.e., non-monotonic reasoning, is a significant open problem in symbolic AI. The formulation of our method is flexible, thus it is possible to incorporate techniques like *belief revision* to deal with this problem in the future.