Efficient Rectification of Neuro-Symbolic Reasoning Inconsistencies by Abductive Reflection (Extended Abstract)*

Wen-Chao Hu[1,2], **Wang-Zhou Dai**[1,2], **Yuan Jiang**[1,2], **Zhi-Hua Zhou**[1,2]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Artificial Intelligence, Nanjing University, China
[3]School of Intelligence Science and Technology, Nanjing University, China
{huwc, daiwz, jiangy, zhouzh}@lamda.nju.edu.cn

Abstract

Neuro-Symbolic (NeSy) AI could be regarded as an analogy to human dual-process cognition, modeling the intuitive System 1 with neural networks and the algorithmic System 2 with symbolic reasoning. However, for complex learning targets, NeSy systems often generate outputs inconsistent with domain knowledge. Inspired by the human Cognitive Reflection, which promptly detects errors in our intuitive response and revises them by invoking the System 2 reasoning, we propose to improve NeSy systems by introducing Abductive Reflection (ABL-Refl) based on the Abductive Learning (ABL) framework. ABL-Refl leverages domain knowledge to abduce a reflection vector during training, which can then flag potential errors in the neural network outputs and invoke abduction to rectify them and generate consistent outputs during inference. Experiments show that ABL-Refl outperforms state-of-the-art NeSy methods, achieving excellent accuracy with fewer training resources and enhanced efficiency.

1 Introduction

Human decision-making is generally recognized as an interaction between two systems: System 1 quickly generates an intuitive response, and System 2 engages in further algorithmic and slow reasoning [Frederick, 2005; Kahneman, 2011]. In Neuro-Symbolic (NeSy) AI, neural networks often resemble System 1 for rapid pattern recognition, and symbolic reasoning mirrors System 2 to leverage domain knowledge and handle complex problems thoughtfully, [Bengio, 2019]. Like human System 1 reasoning, when facing complicated tasks, neural networks often produce unreliable outputs which cause inconsistencies with domain knowledge. These inconsistencies can then be reconciled with the help of the symbolic reasoning counterpart [Hitzler, 2022].

Abductive Learning (ABL) [Zhou, 2019; Zhou and Huang, 2022] is a framework for bridging machine learning and logical reasoning while preserving full expressive power in each side. In ABL, the two components operate in a mutually beneficial loop, continuously improving each other. It features an easy-to-use open-source toolkit [Huang *et al.*, 2024] and has been applied in various practical tasks [Huang *et al.*, 2020; Cai *et al.*, 2021; Wang *et al.*, 2021; Gao *et al.*, 2024]. However, previous implementations of ABL require a highly discrete optimization process to maximize the consistency between the two components, and this optimization has high complexity which encumbers, thereby limiting the efficiency and applicability to large-scale scenarios.

Human reasoning exploits both sides efficiently, a hypothetical model for this process is called Cognitive Reflection, where System 1 quickly generates an approximate over-all solution, and then seamlessly hands complex parts to System 2 [Frederick, 2005]. The key to this process is the reflection mechanism, which promptly detects which part in the intuitive response may contain inconsistencies with domain knowledge and invokes System 2 to rectify them [Sinayev and Peters, 2015]. Following the reflection, the process of the step-by-step formal reasoning becomes less complex: With a largely reduced search space, deriving the correct solution for System 2 becomes straightforward.

Inspired by this phenomenon, we propose a general enhancement, *Abductive Reflection (ABL-Refl)*. Based on ABL framework, ABL-Refl preserves full expressive power of neural networks and symbolic reasoning, while replacing the time-consuming consistency optimization with the reflection mechanism, thereby significantly improves efficiency and applicability. Specifically, in ABL-Refl, a reflection vector is concurrently generated with the neural network intuitive output, which flags potential errors in the output and invokes symbolic reasoning to perform abduction, thereby rectifying these errors and generating a new output that is more consistent with domain knowledge. During model training, the training information for the reflection derives from domain knowledge. In essence, the reflection vector is abduced from domain knowledge and serves as an attention mechanism for narrowing the problem space of symbolic reasoning.

2 Abductive Learning

This section presents problem setting and the *Abductive Learning (ABL)* framework.

The main task of this paper is as follows: The input is raw data $\boldsymbol{x}$, which can be in either symbolic or sub-symbolic

*This is an abridged version of the paper [Hu *et al.*, 2025] that won the Outstanding Paper Award at AAAI 2025.

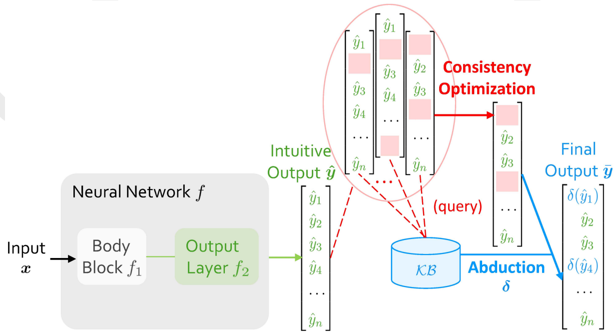

Figure 1: Abductive Learning (ABL) framework.

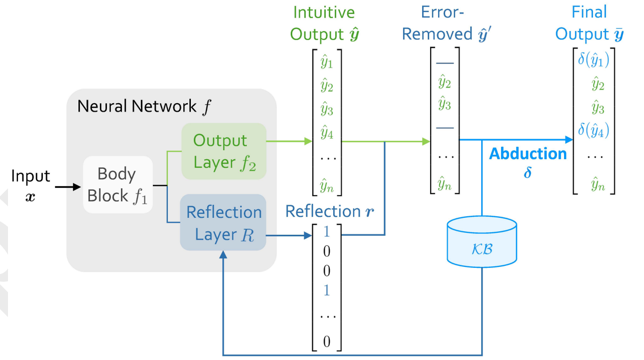

Figure 2: Architecture of Abductive Reflection (ABL-Refl). It replaces the external consistency optimization module with an efficient reflection mechanism, which is abduced directly from $\mathcal{KB}$.

form, and the target output is $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$, with each y_i being a symbol from a set $\mathcal{Y}$ that contains all possible output symbols. We assume two key components at our disposal: neural network f and domain knowledge base $\mathcal{KB}$. f can directly map $\boldsymbol{x}$ to $\boldsymbol{y}$, and $\mathcal{KB}$ holds constraints between the symbols in $\boldsymbol{y}$. $\mathcal{KB}$ can assume various forms, including propositional logic, first-order logic, mathematical or physical equations, etc., and can perform symbolic reasoning operations by exploiting the corresponding symbolic solver. The output $\boldsymbol{y}$ should adhere to the constraints in $\mathcal{KB}$, otherwise it will inevitably contain errors that lead to inconsistencies with the domain knowledge and incorrect reasoning results.

When ABL receives an input $\boldsymbol{x}$, it initially employs f to map $\boldsymbol{x}$ into an intuitive output $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n]$. When f is under-trained, $\hat{\boldsymbol{y}}$ might contain errors leading to inconsistencies with $\mathcal{KB}$. ABL then tries to rectify them, and obtains a revised $\bar{\boldsymbol{y}}$. As shown in Figure 1, the final output, $\bar{\boldsymbol{y}}$, consists of two parts: the green part retains the results from neural network, and the blue part is the modified result obtained by abduction, a basic form of symbolic reasoning that seeks plausible explanations for observations based on $\mathcal{KB}$.

Specifically, the process of obtaining $\bar{\boldsymbol{y}}$ can be divided into two sequential steps. The first step, consistency optimization, determines which positions in $\hat{\boldsymbol{y}}$ include elements that contain errors causing inconsistencies, so that performing abduction at these positions will yield a $\bar{\boldsymbol{y}}$ consistent with $\mathcal{KB}$. Once these positions are determined, the second step is rectifying by abduction, which then becomes easy for $\mathcal{KB}$ and its corresponding symbolic solver.

Challenge. In previous ABL, consistency optimization has always been a computational bottleneck. It operates as an external module using zeroth-order optimization methods [Dai *et al.*, 2019; Zhou and Huang, 2022]. For each time of inference, it involves repetitively selecting various possible positions and querying the $\mathcal{KB}$ to see if a consistent result can be inferred, and the number of such queries required escalates exponentially as data scale increases.

3 Abductive Reflection

To address the challenges above, we propose *Abductive Reflection (ABL-Refl)*, a general enhancement method that incorporates a reflection mechanism into the ABL framework.

3.1 Architecture

Let's first revisit the role of the neural network f when we map the input to symbols from the set $\mathcal{Y}$. Typically, the raw data is first passed through the body block of the network, denoted by f_1, resulting in a high-dimensional embedding which encapsulates a wealth of feature information of the raw data. The result of f_1 is subsequently passed into several layers, usually linear layers, denoted by f_2, to obtain the intuitive output: $\hat{\boldsymbol{y}} = \operatorname{argmax}(f_2(f_1(\boldsymbol{x}))) \in \mathcal{Y}^n$.

Besides the structure described above, as shown in Figure 2, our architecture further incorporates a reflection layer R after the body block f_1, generating a reflection vector: $\boldsymbol{r} = \operatorname{argmax}(R(f_1(\boldsymbol{x}))) \in \{0,1\}^n$. The reflection layer R and reflection vector $\boldsymbol{r}$ together constitute the reflection mechanism. This vector $\boldsymbol{r}$ has the same dimensionality n as the intuitive output $\hat{\boldsymbol{y}}$, and each element, r_i, acts as a binary classifier to indicate whether the corresponding element $\hat{y}_i$ is an error leading to inconsistencies with $\mathcal{KB}$ (flagged as 1 for an error, and 0 otherwise). The reflection vector $\boldsymbol{r}$ is generated concurrently with the intuitive response during inference, resonating with human cognition where cognitive reflection typically forms right upon generation of an intuitive response [Frederick, 2005].

With the initial intuitive output $\hat{\boldsymbol{y}}$ and the corresponding reflection vector $\boldsymbol{r}$, we seamlessly obtain the error-removed output $\hat{\boldsymbol{y}}'$: In $\hat{\boldsymbol{y}}'$, elements flagged as error by $\boldsymbol{r}$ are removed and left as blanks, while the rest are retained. Subsequently, $\mathcal{KB}$ applies abduction to fill in these blanks, thereby generating an output $\bar{\boldsymbol{y}}$ that is consistent with $\mathcal{KB}$. That is:

$$\bar{y}_i = \begin{cases} \hat{y}_i, & r_i = 0 \\ \delta(\hat{y}_i), & r_i = 1 \end{cases} \qquad i = 1, 2, \ldots, n$$

where δ denotes abduction. We treat $\bar{\boldsymbol{y}} = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n]$ as the final output.

During training, the reflection is abduced from $\mathcal{KB}$ by directly leveraging information from it. It can be seen as an attention mechanism generated from neural networks, which quickly focus symbolic reasoning on specific areas, hence largely narrowing the reasoning problem space.

Benefits. Compared to previous ABL implementations, ABL-Refl replaces the zeroth-order consistency optimization

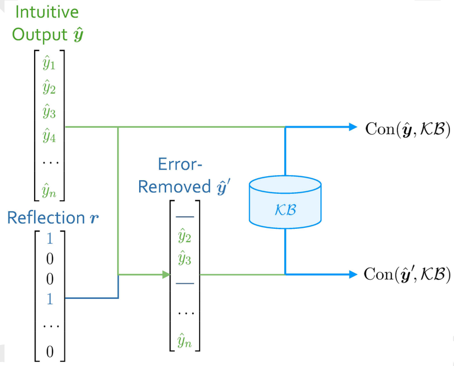

Figure 3: Consistency measurements.

module with the reflection mechanism to address the computational bottleneck. In this way, the need for a substantial number of querying $\mathcal{KB}$ is mitigated: After promptly pinpointing inconsistencies in System 1 output, regardless of the data scale, only a single invocation of $\mathcal{KB}$ is required to obtain a rectified and more consistent output.

Another thing worth noticing is that, in the architecture, the reflection layer directly connects to the body block, which helps leveraging information from the embeddings and linking more closely with the raw data. Therefore, the reflection vector $\boldsymbol{r}$ establishes a more direct and tighter bridge between raw data and domain knowledge.

3.2 Training Paradigm

In ABL-Refl, when each input $\boldsymbol{x}$ is processed by the neural network, we obtain the intuitive output $\hat{\boldsymbol{y}}$ and the reflection vector $\boldsymbol{r}$, and subsequently obtain the error-removed (by $\boldsymbol{r}$) output $\hat{\boldsymbol{y}}'$. With $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{y}}'$, we can measure their consistency with $\mathcal{KB}$. We denote these consistency measurements as $\text{Con}(\hat{\boldsymbol{y}}, \mathcal{KB})$ and $\text{Con}(\hat{\boldsymbol{y}}', \mathcal{KB})$, as shown in Figure 3.

Consequently, the improvement in consistency measurement after reflection, as denoted by

$$\Delta\text{Con}_{\boldsymbol{r}}(\hat{\boldsymbol{y}}) = \text{Con}(\hat{\boldsymbol{y}}', \mathcal{KB}) - \text{Con}(\hat{\boldsymbol{y}}, \mathcal{KB})$$

naturally indicates the effectiveness of the reflection vector: A higher value of it signifies that reflection $\boldsymbol{r}$ can more effectively detect inconsistencies within $\hat{\boldsymbol{y}}$. Our training goal is to guide the neural network's parameters towards generating reflections that can maximize this value. Given that $\Delta\text{Con}_{\boldsymbol{r}}(\hat{\boldsymbol{y}})$ is usually a discrete value, we employ the REINFORCE algorithm to achieve this goal [Williams, 1992], which optimizes the policy (implicitly defined by neural network f) through maximizing a specified reward — in this case, $\Delta\text{Con}_{\boldsymbol{r}}(\hat{\boldsymbol{y}})$. This process leads to the following consistency loss:

$$L_{con}(\boldsymbol{x}) = -\Delta\text{Con}_{\boldsymbol{r}}(\hat{\boldsymbol{y}}) \cdot \nabla_\theta \log f_\theta\left(\hat{\boldsymbol{y}}, \boldsymbol{r} \mid \boldsymbol{x}\right) \tag{1}$$

where θ are parameters of neural network f.

Additionally, given that the time abduction required often escalates with problem size, we want to invoke it judiciously during inference, applying it only when it is truly necessary. Therefore, we aim to avoid the reflection vector from flagging too many elements in $\hat{\boldsymbol{y}}$ as error. To achieve this, we then introduce a reflection size loss:

$$L_{size}(\boldsymbol{x}) = \Phi\left(C - \frac{1}{n}\sum_{i=1}^{n}\left(1 - R\left(f_1(\boldsymbol{x})\right)_i\right)\right) \tag{2}$$

where $\Phi(a) \triangleq \max(0, a)^2$ and C is a hyperparameter.

In addition to the above-mentioned training methods, using labeled data, we employ data-driven supervised training methods similar to common neural network training paradigm. The loss function in this process, e.g., cross-entropy loss, is denoted by $L_{labeled}(\boldsymbol{x}, \boldsymbol{y})$.

Therefore, combining all the training loss, the total loss for ABL-Refl is represented as follows:

$$\begin{aligned}\mathcal{L} = &\frac{1}{|D_l|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in D_l} L_{labeled}(\boldsymbol{x}, \boldsymbol{y}) \\ &+ \frac{1}{|D_l \cup D_u|}\sum_{\boldsymbol{x}\in D_l\cup D_u}\left(\alpha L_{con}(\boldsymbol{x}) + \beta L_{size}(\boldsymbol{x})\right)\end{aligned} \tag{3}$$

where α and β are hyperparameters, D_l and D_u are the labeled and unlabeled datasets, respectively. Note that neither L_{con} nor L_{size}, which are loss functions specifically related to the reflection, incorporate information from the data label. Instead, we leverage training information directly from $\mathcal{KB}$ to train the reflection. Also, despite sharing the prior feature layers, the output layer f_2 and reflection layer R utilize different training information, thereby decoupling the objectives of intuitive problem-solving and inconsistency reflection.

4 Experiments

In this section, we will test our method on the NeSy benchmark task of solving Sudoku to verify its effectiveness. Next, we will change the Sudoku input from symbols to images, which requires integrating and simultaneous reasoning with both sub-symbolic and symbolic elements, representing one of the most challenging tasks in this field.

4.1 Solving Sudoku

Dataset and Setting. This task aims to solve a 9×9 Sudoku: Given 81 digits of 0-9 (where 0 represents a blank space) in a 9×9 board, we aim to find a solution $\boldsymbol{y} \in \{1, 2, \ldots, 9\}^{81}$ that adhere to the Sudoku rules:. We use datasets from a publicly available site [Vopani, 2019].

For the neural network f, we use a simple graph neural network (GNN) as the body block f_1, and then connects to both a linear output layer f_2 to obtain the intuitive output $\hat{\boldsymbol{y}}$ and a linear reflection layer R to obtain the reflection vector $\boldsymbol{r}$. The domain knowledge base $\mathcal{KB}$ contains the Sudoku rules mentioned above. We express $\mathcal{KB}$ in the form of propositional logic and utilize the MiniSAT solver [Sörensson, 2010], an open-source SAT solver, as the symbolic solver.

Compared Methods and Results. We compare ABL-Refl with the following baseline methods: 1) Recurrent Relational Network (RRN) [Palm *et al.*, 2018], a pure neural network method, 2) CL-STE [Yang *et al.*, 2022], a representative method of logic-based regularized loss, and 3) SAT-Net [Wang *et al.*, 2019], a differentiable maximum satisfiability integrated in neural networks.

Method	Training Time (min)	Inference Time (s)	Inference Accuracy
RRN	114.8±7.8	0.19±0.01	73.1±1.2
CL-STE	173.6±9.9	0.19±0.02	76.5±1.8
SATNet	140.3±6.8	0.11±0.01	74.1±0.4
ABL-Refl	**109.8**±**10.8**	0.22±0.02	**97.4**±**0.3**

Table 1: Training time (for a total of 100 epochs), inference time and accuracy on solving Sudoku.

Method	Inference Time (s)	Inference Accuracy
SATNet	0.12±0.01	63.5±2.2
CNN+Solver	0.23±0.02	67.8±4.2
ABL-Refl	0.22±0.02	**77.8**±**5.8**
ABL-Refl (with pretrained CNN)	0.22±0.02	**93.5**±**3.2**

Table 2: Inference time and accuracy on solving visual Sudoku.

We report the training time, inference time and accuracy in Table 1. We may see that our method outperforms the baselines significantly, improving by over 20% while maintaining a comparable inference time. Furthermore, our method reaches high accuracy in only a few epochs, significantly reducing training time. Even considering under identical training epochs, our total training time is less than baseline methods, despite involving a time-consuming symbolic solver.

4.2 Solving Visual Sudoku

Dataset and Setting. In this section, we modify the input from symbolic digits to MNIST images (handwritten digits of 0-9). We use the dataset provided in SATNet.

In order to process image data, we first pass each image through a LeNet convolutional neural network (CNN) [LeCun *et al.*, 1998] to obtain the probability of each digit. The rest of our setting follows from that described in Section 4.1.

Compared Methods and Results. We compare ABL-Refl with SATNet. We report the results in Table 2. Compared to SATNet, ABL-Refl shows notable improvement in reasoning accuracy within only a few training epochs. We then consider pretraining the CNN in advance using self-supervised learning methods [Chen *et al.*, 2020] and find that this can further improve accuracy.

We also compare with CNN+Solver: each image is first mapped to symbolic form by a fully trained CNN (with 99.6% accuracy on the MNIST dataset) and then directly fed into the symbolic solver to fill in the blanks and derive the final output. In such scenarios, the problem space for the symbolic solver includes all the Sudoku blanks, and additionally, since the symbolic solver cannot revise errors from CNN, any inaccuracies in CNN's output could lead the symbolic solver to crash (i.e., output no solution). Consequently, inference accuracy and time are adversely affected.

Finally, an overview of Sections 4.1 and 4.2 also suggests that ABL-Refl is capable of handling both symbolic and subsymbolic forms of input data.

Method	Recall	Inference Time (s)	Inference Accuracy
ABL	Timeout	Timeout	Timeout
NN Confidence	82.64±2.78	0.24±0.03	64.3±6.2
NASR	95.86±0.96	0.26±0.02	82.7±4.4
ABL-Refl	**99.04**±**0.85**	**0.22**±**0.02**	**93.5**±**3.2**

Table 3: Recall, inference time and accuracy. "Timeout" indicates that inference takes more than 1 hour.

5 Effects of Reflection Mechanism

This section analyzes the reflection mechanism in ABL-Refl, which is abduced from domain knowledge and acts as an attention mechanism to guide symbolic reasoning. It plays a central role in identifying neural output errors and invoking symbolic rectification.

To corroborate the effectiveness of the reflection, we conduct direct comparison with other error-detection methods on the solving visual Sudoku task in Section 4.2. We report the recall (the percentage of errors from neural networks that can be identified), inference time and accuracy in Table 3.

(1) **ABL**, minimizing the inconsistency of intuitive output and knowledge base with an external zeroth-order consistency optimization module, as detailed in Section 2. Due to the large data scale (output dimension $n = 81$), the potential rectifications can reach up to 2^{81}, resulting in an overwhelmingly large search space for consistency optimization. Therefore, it takes several hours to complete inference.

(2) **NN Confidence**, retaining intuitive output with the top 80% confidence from the neural network result and passing the remaining into symbolic reasoning. Since the pure data-driven neural network training does not explicitly incorporate $\mathcal{KB}$ information, a low confidence from it does not necessarily indicate an inconsistency with the domain knowledge.

(3) **NASR** [Cornelio *et al.*, 2023], using a Transformer-based external selection module to detect error, and the module is trained on a large synthetic dataset in advance. Our method outperforms it without the need of a synthetic dataset. This may step from the fact that our method can leverage information directly from the body block of neural network, establishing a deeper connection with the raw data.

6 Conclusion

In this paper, we present *Abductive Reflection (ABL-Refl)*. It leverages domain knowledge to abduce a reflection vector, which flags potential errors in neural network outputs and then invokes abduction, serving as an attention mechanism for symbolic reasoning to focus on a much smaller problem space. ABL-Refl preserves the integrity of both machine learning and logical reasoning with superior inference speed and high versatility. Therefore, it has the potential for broad application. In the future, it can be applied to large language models [Mialon *et al.*, 2023] to help identify errors within their outputs, and subsequently exploit symbolic reasoning to enhance their trustworthiness and reliability.

Acknowledgments

This research was supported by the NSFC (62176117, 62206124) and Jiangsu Science Foundation Leading-edge Technology Program (BK20232003).

References

[Bengio, 2019] Yoshua Bengio. From system 1 deep learning to system 2 deep learning. In *Neural Information Processing Systems*, 2019.

[Cai *et al.*, 2021] Le-Wen Cai, Wang-Zhou Dai, Yu-Xuan Huang, Yu-Feng Li, Stephen H Muggleton, and Yuan Jiang. Abductive learning with ground knowledge base. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, pages 1815–1821, 2021.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Cornelio *et al.*, 2023] Cristina Cornelio, Jan Stuehmer, Shell Xu Hu, and Timothy Hospedales. Learning where and when to reason in neuro-symbolic inference. In *The Eleventh International Conference on Learning Representations*, 2023.

[Dai *et al.*, 2019] Wang-Zhou Dai, Qiuling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Frederick, 2005] Shane Frederick. Cognitive reflection and decision making. *Journal of Economic perspectives*, 19(4):25–42, 2005.

[Gao *et al.*, 2024] En-Hao Gao, Yu-Xuan Huang, Wen-Chao Hu, Xin-Hao Zhu, and Wang-Zhou Dai. Knowledge-enhanced historical document segmentation and recognition. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI'24)*, pages 8409–8416, 2024.

[Hitzler, 2022] Pascal Hitzler. *Neuro-symbolic artificial intelligence: The state of the art*. IOS Press, 2022.

[Hu *et al.*, 2025] Wen-Chao Hu, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI'25)*, pages 17333–17341, 2025.

[Huang *et al.*, 2020] Yu-Xuan Huang, Wang-Zhou Dai, Jian Yang, Le-Wen Cai, Shaofen Cheng, Ruizhang Huang, Yu-Feng Li, and Zhi-Hua Zhou. Semi-supervised abductive learning and its application to theft judicial sentencing. In *Proceedings of the 20th IEEE International Conference on Data Mining (ICDM'20)*, pages 1070–1075, 2020.

[Huang *et al.*, 2024] Yu-Xuan Huang, Wen-Chao Hu, En-Hao Gao, and Yuan Jiang. Ablkit: A python toolkit for abductive learning. *Frontiers of Computer Science, pp. to appear*, 2024.

[Kahneman, 2011] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Mialon *et al.*, 2023] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*, 2023.

[Palm *et al.*, 2018] Rasmus Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. *Advances in neural information processing systems*, 31, 2018.

[Sinayev and Peters, 2015] Aleksandr Sinayev and Ellen Peters. Cognitive reflection vs. calculation in decision making. *Frontiers in psychology*, 6:532, 2015.

[Sörensson, 2010] Niklas Sörensson. Minisat 2.2 and minisat++ 1.1. *A short description in SAT Race*, 2010.

[Vopani, 2019] Vopani. 9 million sudoku puzzles and solutions. https://www.kaggle.com/datasets/rohanrao/sudoku, 2019. Accessed: 2024-08-01.

[Wang *et al.*, 2019] Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pages 6545–6554. PMLR, 2019.

[Wang *et al.*, 2021] Jiachen Wang, Dazhen Deng, Xiao Xie, Xinhuan Shu, Yu-Xuan Huang, Le-Wen Cai, Hui Zhang, Min-Ling Zhang, Zhi-Hua Zhou, and Yingcai Wu. Tac-valuer: Knowledge-based stroke evaluation in table tennis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*, pages 3688–3696, 2021.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[Yang *et al.*, 2022] Zhun Yang, Joohyung Lee, and Chiyoun Park. Injecting logical constraints into neural networks via straight-through estimators. In *International Conference on Machine Learning*, pages 25096–25122. PMLR, 2022.

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

[Zhou, 2019] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62:1–3, 2019.