Curriculum Abductive Learning for Mitigating Reasoning Shortcuts

Wen-Da Wei[1,2] , **Xiao-Wen Yang**[1,2] , **Jie-Jing Shao**[1] and **Lan-Zhe Guo**[1,3]
[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
{weiwd, yangxw, shaojj, guolz}@lamda.nju.edu.cn

Abstract

Abductive Learning (ABL), a prominent neural-symbolic learning algorithm, integrates perception models with logical reasoning via intermediate symbolic concepts, substantially improving the interpretability and generalization of AI systems. However, a significant challenge in this domain is the issue of reasoning shortcuts, where the system achieve high final prediction accuracy but generate incorrect intermediate concept inferences, severely undermining ABL's interpretability and generalization capabilities. Current mitigation methods to this problem often neglect potential correlations among training samples, leading to suboptimal performances. This paper innovatively reveals that simple samples can facilitate the learning of intermediate concepts in complex samples, prompting our proposed method Curriculum Abductive Learning (**CurABL**) technique. This approach employs a curriculum training strategy, integrating a knowledge transfer mechanism from simple to complex samples, effectively addressing the issue of reasoning shortcuts. Comprehensive experimental results demonstrate that the **CurABL** method substantially improves the ABL framework's capability to extract intermediate concepts especially in difficult tasks and accelerates the training convergence rate, thus markedly enhancing its robustness against reasoning shortcuts.

1 Introduction

Abductive Learning (ABL) [Zhou, 2019; Cai *et al.*, 2021; Dai *et al.*, 2019] as a novel and flexible neural-Symbolic (NeSy) framework has received significant attention recently [He *et al.*, 2024; Shao *et al.*, 2025; Jia *et al.*, 2025] for its effective integration of data-driven machine learning and knowledge-driven symbolic reasoning. Within the framework of ABL, neural networks [LeCun *et al.*, 2015] serve as the perception model extract symbolic concepts with practical meaning from raw inputs (e.g., images or text), and the symbolic Knowledge base KB utilizes its logical reasoning capabilities to infer the final target label based on the intermediate symbolic concepts. While this is similar to most NeSy frameworks [Xu *et al.*, 2018; Badreddine *et al.*, 2022; Manhaeve *et al.*, 2018], what sets ABL apart is that it does not attempt to make symbolic knowledge differentiable. Instead, it leverages abductive reasoning to construct pseudo-labels for the intermediate symbolic concepts, which are used for updating the machine learning model. Benefiting from the combination of the interpretability of symbolic knowledge with the learning capabilities of neural networks, abductive learning exhibit enhanced potential for interpretability, generalization, and adaptability to new tasks. This potential is largely attributed to the **successfully learning of high-quality intermediate concepts**, which serve as a foundation for their strengths.

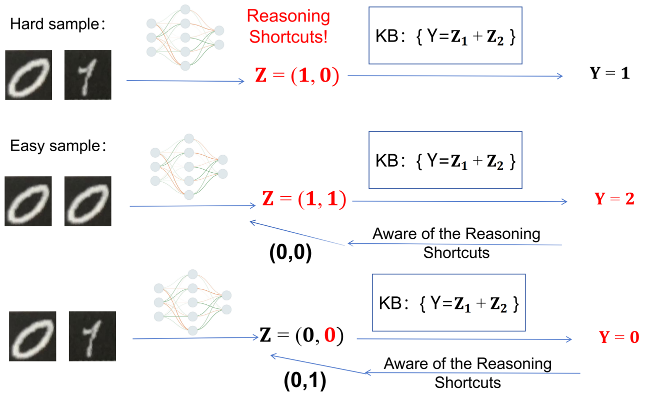


Figure 1: Reasoning shortcuts example and curriculum mechanism.

Nevertheless, existing research [Marconato *et al.*, 2023b] has emphasized that the ABL framework is highly vulnerable to *reasoning shortcuts*, which is a widespread problem also affecting a variety of other typical NeSy algorithms. It refers to the phenomenon where, through training, ABL can attain high accuracy in predicting the final target label while with the **incorrect intermediate concepts**, which severely undermines its interpretability, generalization, and other core advantages. For example, according to the first example in Figure 1, the perception model acquires a reasoning shortcut by confusing the digits 0 and 1 yet this does not affect the prediction of the final label. Several analytical works [Marconato *et al.*, 2023b; Yang *et al.*, 2024] on reasoning shortcuts have been proposed to investigate its causes and quantify its harm. They collectively point out that the fundamental cause of reasoning shortcuts lies in the insufficient constraint imposed by the knowledge base on the intermediate concepts. In particular, during the training process, the knowledge base will generate multiple pseudo-labels for the samples, only one of which is truly correct. As a result, the perception model receives inexact supervision, which hinders its ability to accurately extract the correct intermediate concepts. Although some mitigating methods such as incorporating concept-supervised data [Huang *et al.*, 2020], smoothing labels [Müller *et al.*, 2019] and Bears [Marconato *et al.*, 2024] have been proposed, they either introduce additional information or adopt compromise strategies, failing to fundamentally address reasoning shortcuts.

However, we notice that all previous works have overlooked an important mechanism in ABL training: for a given task, both simple and difficult samples exist, and the simple samples seem to facilitate the learning of the difficult ones–an aspect that we first to uncover. We will explain and analyze this mechanism in detail in Section 4.1. This phenomenon seems to play a significant role in fundamentally mitigating the reasoning shortcuts, providing new insights into addressing this widespread issue.

Inspired by the significant finding, we further observe that the curriculum learning paradigm can be seamlessly utilized to address the issue of reasoning shortcuts. Curriculum learning [Bengio *et al.*, 2009] involves presenting samples to a model in a meaningful order, starting with simpler tasks and gradually increasing complexity, enabling the model to learn the easy concepts sooner and the advanced concepts later, thus improving its accuracy, convergence speed, and stability [Soviany *et al.*, 2022]. Therefore, in this paper we propose a highly innovative, effective algorithm, Curriculum Abductive Learning (**CurABL**) to alleviate reasoning shortcuts: incorporating curriculum learning paradigm into the ABL training process. Specifically, our algorithm consists of two main components: (i) we first design a precise measure to evaluate the complexity of the samples by considering the relationships between samples to construct a graph and leveraging the information from this graph to assess sample complexity; (ii) we then construct a curriculum learning training framework based on the information derived from the graph to train the ABL model. We empirically evaluate our algorithm across various datasets, validating its effectiveness in alleviating reasoning shortcuts in ABL framework and also increasing the convergence speed of its training process.

We summarize the contributions of our work:

(i)We are the first to identify a key mechanism in the training process of ABL: simple samples play an effective role in facilitating ABL learning from difficult samples, which offers significant insights for the ABL framework and the issue of reasoning shortcuts.

(ii)We propose a highly novel and ingenious algorithm Curriculum Abductive Learning (**CurABL**) to mitigating the reasoning shortcuts problem fundamentally without any additional information: incorporating the curriculum learning paradigm into the ABL training process to mitigating the reasoning shortcuts problem fundamentally.

(iii)Through extensive experiments, we valid that our algorithm significantly enhances the ability of ABL to resist reasoning shortcuts on challenging datasets, while also offering the benefit of improving the training convergence rate.

2 Related Work

Neural-symbolic Learning Neural-symbolic learning paradigm [Besold *et al.*, 2021; Raedt *et al.*, 2020] seeks to integrate neural networks with symbolic reasoning, in order to achieve a more comprehensive form of Artificial Intelligence. Typical methods [Yang *et al.*, 2022; Xu *et al.*, 2018; Fischer *et al.*, 2019; Huang *et al.*, 2021a] propose neural-symbolic learning approaches to incorporate symbolic rules as logic constraints, ensuring that their outputs strictly adhere to the rules. Furthermore, several techniques [Badreddine *et al.*, 2022; Manhaeve *et al.*, 2018] have specifically focused on integrating neural networks with established tools for logical reasoning. However, regardless of the perspective from which neural-symbolic methods are designed, they are prone to reasoning shortcuts.

Abductive learning [Zhou, 2019] is a novel framework in the field of neural-symbolic. The primary focus of this approach is to handle the discrete intermediate symbolic concepts, which act as pseudo-labels to upadate the model during the learning process and as variables for abductive reasoning. Several variants of ABL have been proposed to optimize the framework's abductive process or adapt to different settings. Cai *et al.* [2021] extend the ABL framework by utilizing a logical domain knowledge base, represented through groundings. Huang *et al.* [2021b] employs a similarity-based consistency metric to determine the most suitable pseudo-label among all possible abduction results, thereby improving the optimization process of the ABL framework in speed and stability. In the semi-supervised setting, Huang *et al.* [2020] applied the ABL framework to address the theft judicial sentencing problem. Similarly, reasoning shortcuts severely undermine the advantages and performance of ABL.

Reasoning Shortcuts Reasoning shortcuts is a significant and challenging issue affecting ABL and most other neural-symbolic frameworks problematically. Wang *et al.* [2023] pointed out the existence of weak supervision on intermediate concepts in neuro-symbolic systems and Marconato *et al.* [2023a] formally introduced the concept of reasoning shortcuts. Several strategies have been proposed to mitigate the reasoning shortcuts. Li *et al.* [2023] propose a minimax objective that ensures the concepts learned by the model satisfy the knowledge base and have fewer shortcuts. Manhaeve *et al.* [2018] introduced the use of a pre-trained model, and Bears [Marconato *et al.*, 2024] utilizes ensemble techniques to enhance the NeSy algorithm's ability to identify shortcuts. Besides, smoothing labels [Müller *et al.*, 2019] is also a concise trade-off strategy to prevent reasoning shortcuts from becoming overly severe. However, the aforementioned methods either require additional information or adopt compromise strategies, failing to fundamentally resolve reasoning shortcuts.

Furthermore, there are also some theoretical analyses on reasoning shortcuts. Marconato *et al.* [2023b] formally defined the reasoning shortcuts problem as representative, theoretically quantified its harm using permutations and analyzed the causes of reasoning shortcuts. Yang *et al.* [2024] proposed a very critical concept D_{KB}, the complexity of the knowledge base, which is closely related to reasoning shortcuts, and established a theoretical framework quantifying the severity of reasoning shortcuts. Both works identified that the root cause of reasoning shortcuts lies in the insufficient constraints imposed by the knowledge base on intermediate concepts, providing valuable insights and strong guidance. They also concluded that the reasoning shortcuts is a challenging problem to address. However, both works fail to consider the critical factor we have uncovered, resulting in a discrepancy between their theoretical quantification of the reasoning shortcuts in a task and the actual in practice.

Curriculum Learning Paradigm Curriculum learning is a classic machine learning paradigm that trains models in a meaningful order, progressing from easy samples to harder ones [Bengio *et al.*, 2009]. The paradigm can achieve an increase of the convergence speed of the training process and a better accuracy over the standard training approach based on random data shuffling [Soviany *et al.*, 2022]. Curriculum learning strategies have already been adopted in various application domains, such as weakly supervised object localization [Ionescu *et al.*, 2016], object detection [Sangineto *et al.*, 2019], and neural machine translation [Wang *et al.*, 2019] among many others. However, despite its success in certain domains, curriculum learning has not been adopted in mainstream works. This is because, in most tasks, it is difficult to define what "easy examples" mean. In the problem setting of our paper, defining the complexity of a sample is explicit, and straightforward. Therefore, the curriculum learning paradigm can be seamlessly leveraged by us to address the reasoning shortcuts problem, which is a highly novel and practical idea.

3 Preliminaries and Problem Setting

In this section, we will provide a concise overview of the ABL system and the problem setting of our proposed algorithm. Abductive learning consists of a perception model denoted as f and a symbolic knowledge base denoted as KB. The perception model $f : \mathcal{X} \to \mathcal{Z}$, typically implemented with a neural network, maps the raw input $\mathbf{x} \in \mathcal{X}$ into the intermediate concepts $\mathbf{z} \in \mathcal{Z}$, where $\mathcal{Z}$ is a finite discrete symbol space. The intermediate concepts $\mathbf{z}$, which take on a finite number of values, have precise and human-understandable meanings. We define $\mathcal{Z}$ as a k–dimensional vector space, indicating that there are k types of intermediate concepts, each taking discrete positive integer values. The knowledge base KB consists of a set of logical rules provided by experts, which can infer the final target label $y \in \mathcal{Y}$ through the intermediate concepts $\mathbf{z}$, satisfying that $\mathbf{z}, \text{KB} \models y$.

In the ABL training setting, we are given the training set $S = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$, where the samples are independently and identically distributed (i.i.d.). More formally, we define a joint distribution $\mathcal{P}$ on the space $\mathcal{X} \times \mathcal{Y}$, and S is sampled from distribution $\mathcal{P}$. It is worth noting that within the ABL framework, there are no definitive groundings for the intermediate concept $\mathbf{z}$. During the training process, for each sample in $\mathcal{S}$, the knowledge base KB will receive the estimated concepts $\hat{\mathbf{z}} = f(\mathbf{x})$ and verify whether $\hat{\mathbf{z}}, \text{KB} \models y$. If inconsistent, KB will generate pseudo-labels $\bar{\mathbf{z}}$ satisfying that $\bar{\mathbf{z}}, \text{KB} \models y$ through abductive reasoning, which are used for updating the model f.

However, as we mentioned earlier, due to the insufficient constraints imposed by KB, the same final label can be reasoned by KB from multiple intermediate concepts, resulting in the generation of multiple pseudo-labels for each sample, only one of which is truly correct. This is the fundamental reason for the emergence of reasoning shortcuts. Therefore, ABL must rely on specifically designed metrics (e.g., random selection or Hamming distance [Dai *et al.*, 2019]) to select the best pseudo-label from the set of all pseudo-label candidates, which is often inaccurate, especially in challenging tasks. To formalize this, we define $C(\mathbf{x}, y) = \{\bar{\mathbf{z}} | \bar{\mathbf{z}}, \text{KB} \models y\}$ as the candidate set of all pseudo-labels for a sample $(\mathbf{x}, y)$. The function $Dis : C(\mathbf{x}, y) \to \bar{\mathbf{z}}$ maps the best pseudo-label from the candidate set $C(\mathbf{x}, y)$, selected by the ABL framework using certain strategy. The optimization objective of Abductive Learning can therefore be formalized as:

$$\min_{f} \sum_{(\mathbf{x},y)\in S} \mathcal{L}(f(\mathbf{x}), Dis(C(\mathbf{x}, y)))$$

where $\mathcal{L}$ represents the cross-entropy loss. In our proposed algorithm, the cardinality of the set can be used to $C(\mathbf{x}, y)$ measure the complexity of the sample $C(\mathbf{x}, y)$, providing a both simple and accurate strategy. A larger cardinality indicates that the sample is more complex.

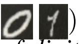
4 Curriculum Abductive Learning

In this section, we provide a detailed introduction to our proposed algorithm Curriculum Abductive Learning **CurABL**, which seeks to fundamentally alleviate the reasoning shortcuts problem by integrating the curriculum learning paradigm into the ABL framework. In Subsection 4.1, we elaborate on the key mechanism in ABL training which we are the first to uncover. Subsection 4.2 details our method for measuring each sample's complexity. In Subsection 4.3, we will then introduce how we construct the curriculum learning training framework.

4.1 Simple Samples help Hard Samples

Reasoning shortcuts hinder the perception model's ability to effectively learn and extract the correct intermediate concepts. However, experimental results show that in some simple tasks such as MNIST-Addition [Manhaeve *et al.*, 2018], despite most samples in the training set having multiple pseudo-labels, the ABL framework is sometimes able to resist the reasoning shortcuts during training. Inspired by this phenomenon, we uncover a highly critical mechanic we mentioned before: for a given task, both simple and difficult samples exist, and the simple samples play a role in facilitating the learning of the difficult ones.

Specifically, simple samples, which are associated with few pseudo-labels, can assist the perception model f in learning from difficult samples with more pseudo-labels when the simple and difficult samples share a subset of intermediate concepts. This is because simpler samples can supervise the perception model to acquire the ability to accurately extract the corresponding intermediate concepts. Once the perception model has successfully learned these concepts, it can implicitly eliminate the incorrect pseudo-labels of difficult samples with partially shared intermediate concepts during their learning process, thereby reducing their complexity. Thus, simpler samples play an effective role in facilitating the perception model learning from difficult samples. We provide an intuitive example below to explain this critical mechanism more clearly.

Example 1. *As illustrated in Figure 1, consider two samples in the MNIST-Addition task:* $(\mathbf{x}_1 = ($ $), y_1 = 0)$ *and* $(\mathbf{x}_2 = ($$), y_2 = 1)$ *sharing the common intermediate concept of digit 0. The first sample is very simple because it contains only one pseudo-label* $(0, 0)$ *for the intermediate concept. The second sample is relatively more difficult because it contains two pseudo-labels* $(0, 1)$ *and* $(1, 0)$ *for the intermediate concept, both of which do not conflict with the knowledge base KB while only* $(0, 1)$ *is correct. The perception model can accurately learn the ability to extract the intermediate concept digit 0 thanks to the first sample, which can implicitly reduce the interference of (1,0) on the second sample and eliminate the incorrect pseudo-label (1,0) for the second sample. Therefore, it can be inferred that the sample* $(\mathbf{x}_1 = ($ $), y_1 = 0)$ *reduces the complexity of the sample* $(\mathbf{x}_2 = ($ $), y_2 = 1)$ *implicitly and facilitate the perception model's learning effect for the second sample.*

The example intuitively and meticulously demonstrates how simpler samples help facilitate the perception model learning from difficult samples. Such sample interaction can create a cascading effect among samples during the process of training, which helps the ABL framework mitigate the reasoning shortcuts.

4.2 Complexity Measurer

Through the aforementioned critical mechanism, simple samples reduce the learning difficulty of difficult samples and facilitate the perception model's learning from them, which helps the ABL framework potentially address reasoning shortcuts. While since this effect is implicit, the mechanism's impact is neither strong enough nor stable. To enhance the effect, we attempt to propose an algorithm to introduce the curriculum learning paradigm into the ABL training process, referred to as Curriculum Abductive Learning (**CurABL**), which can explicitly leverage the positive impact of this mechanism.

To implement the algorithm, designing a measurer to evaluate the complexity of each sample is essential and crucial. An intuitive method to measure the difficulty of a sample $(\mathbf{x}, y)$ is to use the cardinality of its candidate set of all pseudo-labels, $|C(\mathbf{x}, y)|$. However, this approach is less precise, as it fails to consider the effect of simple samples in implicitly reducing the complexity of difficult samples that share partially identical intermediate concepts. Therefore, it is necessary to explore the relationships between samples in the training set and construct a more accurate metric for measuring sample complexity based on the relationships. Specifically, we can construct the relationships between samples that share the same intermediate concepts in the form of a graph and explicitly carry out the incorrect pseudo-label removal process. Subsequently, we can directly use the cardinality of the updated pseudo-label candidate set of the sample after removal to measure the complexity, which maximizes the effectiveness of this mechanism.

How can we identify which samples share the same intermediate concepts? Since there are no definitive groundings for the intermediate concepts $\mathbf{z}$ in the training set, it is challenging without additional information. Fortunately, we have discovered that after training for ABL, the perception model f tends to have a clustering capability. While it can not map inputs to the correct intermediate concepts due to reasoning shortcuts, it consistently maps inputs with the same intermediate concept components to similar embeddings. Specifically, the encoder part E of the perception model f maps the input $\mathbf{x}$ into k embedding spaces, and $E_i(\mathbf{x})$ corresponds to the embedding of the i-th intermediate concept encoded by E. For two inputs $\mathbf{x}_1$ and $\mathbf{x}_2$, if $E_i(\mathbf{x}_1)$ and $E_i(\mathbf{x}_2)$ are sufficiently similar, we can infer that their i-th intermediate concepts are identical. We will validate this observation through experiments presented in Subsection 5.1.

Based on this discovery, we can construct the relationships between samples using a "cold start" approach. We first train the ABL model on the training set S, and then leverage the clustering capability of the encoder E in perception model to identify whether two samples share the same intermediate concepts. We use cosine similarity to measure the similarity of their embeddings. For any two samples $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$, If the the cosine similarity $\frac{E_i^T(\mathbf{x}_1)E_i(\mathbf{x}_2)}{\|E_i(\mathbf{x}_1)\|\|E_i(\mathbf{x}_2)\|} \geq \tau$ exceeds a predefined threshold τ, we infer that that the i-th intermediate concept of the two samples is the same. We use k undirected graphs to store the relationship information between samples, where each graph has N vertices corresponding to the N samples in the training set S. If the i-th intermediate concept of two samples is the same, an edge is added between the two corresponding vertices in the i-th graph. The algorithm 1 provides a clear description of the graph construction process.

Since we have already constructed the relationships between samples related to intermediate concepts in the form of graph-based data structures, the next step is to design a graph-based algorithm to explicitly implement the mechanism where simple samples assist difficult samples by removing incorrect pseudo-labels from their candidate sets. First, we compute and store the initial intermediate concept pseudo-label candidate set $C(\mathbf{x}, y)$ for each sample $(\mathbf{x}, y)$. For simplicity, we use C_j to denote the pseudo-label candidate set of the j-th sample. The goal of our algorithm is to update and refine the set C_j based on the information provided by the graph structure. The update process is based on the following principle: if there exists an edge between the p-th and q-th samples in the i-th graph (where i is a positive integer less than or equal to k), it indicates that the ground truth of the i-th intermediate concept of the p-th sample is identical

Algorithm 1 Cold Start Graph Construction Algorithm

```
Input: Training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, encoder $E$, threshold $\tau$
Parameter: Number of intermediate concepts $k$
Output: $k$ undirected graphs $\{G_1, G_2, \ldots, G_k\}$
Initialize $k$ undirected graphs $\{G_1, G_2, \ldots, G_k\}$, each with $N$ vertices corresponding to the samples in $S$.
for $i = 1$ to $k$ do
    for each pair of samples $(\mathbf{x}_a, y_a)$ and $(\mathbf{x}_b, y_b)$ in $S$ do
        $$\text{similarity} = \frac{E_i^T(\mathbf{x}_a)E_i(\mathbf{x}_b)}{\|E_i(\mathbf{x}_a)\|\|E_i(\mathbf{x}_b)\|}$$
        if similarity $\geq \tau$ then
            Add an edge between vertices $a$ and $b$ in graph $G_i$.
        end if
    end for
end for
return $\{G_1, G_2, \ldots, G_k\}$
```

Algorithm 2 Incorrect Pseudo-Label Removal Algorithm

```
Input: $k$ undirected graphs $\{G_1, G_2, \ldots, G_k\}$, initial pseudo-label candidate sets $\{C_1, C_2, \ldots, C_N\}$ for $N$ samples
Parameter: Number of intermediate concepts $k$
Output: Updated pseudo-label candidate sets $\{C_1, C_2, \ldots, C_N\}$
for $i = 1$ to $k$ do
    for each edge $(p, q)$ in graph $G_i$ do
        for each element $\bar{\mathbf{z}} \in C_p$ do
            if $\bar{\mathbf{z}}_i$ does not match the $i$-th component of any element in $C_q$ then
                Remove $\bar{\mathbf{z}}$ from $C_p$
            end if
        end for
        for each element $\bar{\mathbf{z}} \in C_q$ do
            if $\bar{\mathbf{z}}_i$ does not match the $i$-th component of any element in $C_p$ then
                Remove $\bar{\mathbf{z}}$ from $C_q$
            end if
        end for
    end for
end for
return Updated pseudo-label candidate sets $\{C_1, C_2, \ldots, C_N\}$
```

to that of the q-th sample. Therefore, if the i-th component element $\bar{\mathbf{z}}_i$ of $\bar{\mathbf{z}} \in C_p$ does not match the i-th component of any element in C_q, we can confidently determine that $\bar{\mathbf{z}}$ is an incorrect pseudo-label for the p-th sample and remove it from C_p. The same operation applies to the q-th sample. Based on this principle, we only need to iterate over the edges in the graph and perform the corresponding removal and update operations. The algorithm 2 provides a complete and detailed description of this process.

Using this algorithm, we obtain the updated pseudo-label candidate set C_j for j-th sample, and we use its cardinality $|C_j|$ to measure its complexity. The algorithm not only helps us accurately measure the difficulty of each sample but also explicitly realizes the assistance of simple samples to difficult ones, thereby maximizing the reduction of the difficulty of challenging samples and mitigating the reasoning shortcuts.

4.3 Curriculum Training

Based on the aforementioned measurer, we rank the samples from easy to difficult and then train the ABL model in batches. Notably, during the training process, it is no longer necessary for the KB to generate pseudo-labels. Instead, the pseudo-labels can be directly selected from the already updated pseudo-label candidate set. The curriculum Abductive Learning (**CurABL**) overflow is detailed in Algorithm 3. In conclusion, the CurABL algorithm explicitly implements the mechanism where simple samples assist difficult samples, incorporating a curriculum learning paradigm to train the ABL model. This approach has the potential to mitigate reasoning shortcuts and achieve an increase of the convergence speed of the training process.

Complexity Analysis We analyze the additional overhead of computation and space in our CurABL algorithm. The main components contributing to the overhead are the Graph Construction and Incorrect Pseudo-Label Removal steps. In terms of the time complexity, CurABL introduces a time complexity of $\mathcal{O}(kN^2)$ for graph construction and $\mathcal{O}(kN^C)$ for incorrect pseudo-label removal, where C is the size of the concept space. The space complexity is $\mathcal{O}(kN^2)$ due to the storage of k graphs. The storage of pseudo-label candidate sets does not introduce additional memory overhead, as these sets are also required in the original ABL framework.

5 Experiments

In this section, we conduct experiments to verify our claims and validate the superior performance of **CurABL**. In Subsection 5.1, we verify the clustering capability of the perception model of ABL after training on dataset *MNIST-Additon*. In Subsection 5.2, we describe the experimental setup. In Subsection 5.3, we evaluate the effectiveness of **CurABL** on two datasets, *MNIST-Additon* and *Handwritten Formula Recognition*, by comparing it with different ABL methods.

5.1 Clustering Capability of Perception Model

In this subsection, we attempt to validate the clustering capability of the perception model, demonstrating its ability to map inputs with the same intermediate concept components to similar embeddings. This capability serves as the foundation of our proposed method. We employ the *MNIST-Even-Odd* [Manhaeve *et al.*, 2018] dataset to assess this capability; the specific settings of the dataset will be detailed in Subsection5.2. The dataset *MNIST-Even-Odd* is significantly more susceptible to reasoning shortcuts compared to the original or other MNIST-Addition task, making it a compelling choice for validating the clustering capability of the perception model. To evaluate the model's performance, we present

Algorithm 3 CurABL Training Overflow

Input: Training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, updated pseudo-label candidate sets $\{C_1, C_2, \ldots, C_N\}$, batch size B, number of epochs E
Parameter: Model f, learning rate η
Output: Trained ABL model f

```
Sort $S$ based on Complexity Measurer, from easy to difficult.
for $e = 1$ to $E$ do
    for each batch $D_i \subset S$ with size $B$ do
        Compute batch loss:
        $\mathcal{L}_i = \sum_{(\mathbf{x},y) \in D_i} \mathcal{L}(f(\mathbf{x}), \text{Dis}(C(\mathbf{x}, y)))$
        Update model parameters:
        $f \leftarrow f - \eta \nabla_f \mathcal{L}_i$
    end for
end for
return Trained ABL model $f$
```

the classification results of the perception model using a confusion matrix. The results are depicted in the first image of Figure 2.

According to the figure, we find that due to the profound influence of reasoning shortcuts, the perception model achieves an accuracy of less than 5% for predicting each intermediate digit concept. However, each digit is consistently misclassified into the same intermediate concept. This observation validates our hypothesis regarding the clustering capability of the perception model, providing a strong foundation for the Graph Construction Algorithm.

5.2 Experimental Setup

Settings of MNIST-Addition The MNIST-Addition task [Manhaeve *et al.*, 2018] takes two images of handwritten digits as input and outputs their sum. The dataset is one of the most classic benchmarks in the neuro-symbolic domain, containing a total of 30000 samples. Bears [Manhaeve *et al.*, 2018] introduced the variant of MNIST-Addition task *MNIST-Even-Odd*, where all digits are significantly affected by reasoning shortcuts. The dataset is highly susceptible to Reasoning shortcuts, making it an ideal benchmark for evaluating the effectiveness of mitigating algorithms. However, since there are no inherently simple samples in the *MNIST-Even-Odd* dataset, our approach struggles to take effect. Therefore, we inject 10 samples with unique pseudo-label into the *MNIST-Even-Odd* dataset to better evaluate the effectiveness of our method.

Settings of Handwritten Formula Recognition Additionally, we conduct experiments on the Handwritten Formula Recognition *HWF* task [Li *et al.*, 2020]. In this task, the input is a formula composed of multiple handwritten images, where the length of the formula corresponds to the number of images, and the output is the computed result of the formula. Each image represents an intermediate concept, which can be one of the digits 1–9 or one of the four operators "+","-","*","/", resulting in a total of 13 classes. The output is a real number. The dataset contains $10,000$ samples, with equations of varying lengths in the set $\{1, 3, 5, 7\}$. To further increase the task's difficulty and better evaluate the mitigation effectiveness of our algorithm, we constructed two more challenging datasets, named *HWF-M* and *HWF-H*, by extracting a subset of difficult samples from the HWF dataset. Specifically, we selected all samples of length 7 from the HWF dataset in a total of 6,000 samples. From this subset, we removed samples where the cardinality of the candidate set of all pseudo-labels was less than 10 to create the *HWF-M* dataset. Similarly, we removed samples with a cardinality less than 50 to construct the *HWF-H* dataset. Both datasets contain nearly $6,000$ samples.

Method	**MNIST-Addition**		**MNIST-Even-Odd**	
	Z-ACC	Y-ACC	Z-ACC	Y-ACC
ABL	$0.73_{\pm 0.05}$	$0.63_{\pm 0.08}$	$0.17_{\pm 0.00}$	$0.14_{\pm 0.00}$
ABL-Hamming	$0.80_{\pm 0.16}$	$0.88_{\pm 0.04}$	$0.01_{\pm 0.00}$	$\mathbf{0.94_{\pm 0.00}}$
CurABL	$\mathbf{0.99_{\pm 0.00}}$	$\mathbf{0.98_{\pm 0.00}}$	$\mathbf{0.49_{\pm 0.01}}$	$0.89_{\pm 0.00}$

Table 1: Accuracy on MNIST-Addition and MNIST-Even-Odd

Comparison Methods

We compare our **CurABL** with two baseline methods, ABL and ABL-Hamming [Dai *et al.*, 2019], Here, ABL refers to the original method, which randomly selects the pseudo-label from all pseudo-label candidates. Our experiments focus on evaluating and validating two key advantages of our algorithm: (1) significantly mitigating reasoning shortcuts , and (2) increasing the convergence speed during training. Specifically, we conduct experiments on CurABL, ABL, and ABL-Hamming using the *MNIST-Addition* and *MNIST-Even-Odd* datasets to evaluate the effectiveness of CurABL in mitigating Reasoning shortcuts. Additionally, we compare CurABL and ABL-Hamming on the *HWF*, *HWF-M*, and *HWF-H* datasets to further assess its ability. ABL is omitted from the HWF experiments due to the overwhelming number of pseudo-label candidates, which prevents ABL from converging on this task. This further demonstrates the clear advantages of CurABL over ABL. To evaluate the improvement in training convergence rate, we conduct experiments on CurABL and ABL-Hamming using the *HWF* dataset.

Experimental Details This paragraph provides a detailed explanation of the implementation of the experiments. Due to the relatively low difficulty of the original *MNIST-Addition* dataset, it is unnecessary to utilize our precisely designed Complexity Measurer to sort the samples. While for the *MNIST-Even-Odd* dataset, which is highly sensitive to reasoning shortcuts, we need to employ the accurate Complexity Measurer to sort the samples through the Cold Start Graph Construction Algorithm and the Incorrect Pseudo-Label Removal Algorithm. In the cold start method, we first train the ABL model on the dataset for six epochs and we set the threshold $\tau = 0.95$. To ensure reliability of our results, all experiments are repeated five times with different random seeds. All experiments are implemented by Pytorch and are conducted on an NVIDIA RTX 3090 GPU.

5.3 Empirical Analysis

In this Subsection, we will present our experimental results and analyze them to demonstrate the superior performance

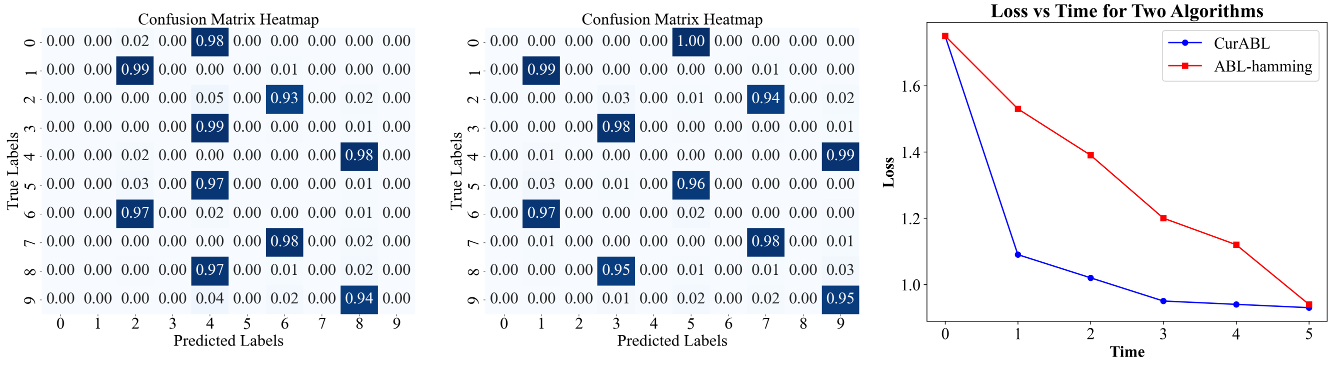


Figure 2: Confusion matrix of intermediate concepts and loss convergence curve

Method	HWF		HWF-M		HWF-H	
	Z-ACC	Y-ACC	Z-ACC	Y-ACC	Z-ACC	Y-ACC
ABL-Hamming	$0.994_{\pm 0.00}$	$0.967_{\pm 0.00}$	$0.986_{\pm 0.00}$	$0.924_{\pm 0.00}$	$0.647_{\pm 0.03}$	$0.390_{\pm 0.05}$
CurABL	$\mathbf{0.998_{\pm 0.00}}$	$\mathbf{0.989_{\pm 0.00}}$	$\mathbf{0.992_{\pm 0.00}}$	$\mathbf{0.953_{\pm 0.00}}$	$\mathbf{0.931_{\pm 0.01}}$	$\mathbf{0.677_{\pm 0.01}}$

Table 2: Accuracy on HWF, HWF-M, and HWF-H

of our proposed algorithm **CurABL**. We showcase the advantages of **CurABL** in two aspects: mitigating reasoning shortcuts and increasing the convergence speed during training. To evaluate the algorithm's ability to mitigate RS, we use the accuracy of the perception model in extracting intermediate concepts on the test set as the evaluation metric. We conducted the experiments as described in the Comparison Methods, and the results are presented in Table 1 and Table 2 and the third image of Figure 2.

In the tables, Z-ACC represents the accuracy of intermediate concept extraction within the model and Y-ACC denotes the accuracy of final label prediction. Each value in the tables corresponds to the mean accuracy, with the variance displayed in the lower-right corner. The highest accuracy for each metric is highlighted in bold. The results show that our method achieves consistent improvements across all datasets and demonstrates significant advantages over the baselines on challenging dataset tasks, validating its effectiveness in mitigating reasoning shortcuts. **CurABL** also exhibit very low variance in performance, highlighting the stability advantage brought by curriculum learning. What's more, we conduct experiments to evaluate its improvement in convergence speed. As shown in the third image of Figure 2, the training loss of the model using our **CurABL** method decreases rapidly compared to the ABL-Hamming method.

Specifically, on the challenging *MNIST-Even-Odd dataset*, we observed that the original ABL method struggles to learn effectively. The ABL-Hamming method demonstrates high accuracy in final label prediction, but fails almost entirely in extracting intermediate concepts correctly. This phenomenon is also clearly illustrated in Subsecton 5.1. In contrast, our **CurABL** method achieves a similar level of performance in final label prediction compared to ABL-Hamming but significantly improves the accuracy of intermediate concept extraction, reaching 50%, thus demonstrating a substantial mitigation effect. To further investigate the mechanisim, we also use a confusion matrix to present the classification results of the perception model trained with **CurABL**, as shown in the second image of Figure 2. We observe that the perception model achieves high accuracy in predicting odd numbers but performs poorly on even numbers, which is an intriguing phenomenon that we plan to explore further in future work.

6 Conclusion

In this paper, we introduced Curriculum Abductive Learning (**CurABL**), a novel and effective algorithm designed to address the pervasive issue of reasoning shortcuts in Abductive Learning. Through our study, we uncovered a key mechanism in the ABL training process: simple samples play a crucial role in facilitating learning from difficult samples by reducing their complexity. Building upon this insight, we proposed a curriculum learning paradigm tailored for ABL, which ranks samples by complexity and processes them in order, seamlessly integrating the positive effects of simple-to-complex learning into ABL training. Extensive experimental results demonstrated the significant advantages of CurABL across various benchmark datasets. Our work also provides key insights into the dynamics between simple and complex samples in neural-symbolic frameworks like ABL. However, our method becomes ineffective on datasets where the differences in sample difficulty are minimal, as the assistance of simple samples to difficult ones cannot be effectively utilized in such training sets.

Acknowledgments

This research was supported by the Key Program of Jiangsu Science Foundation (BK20243012), Leading-edge Technology Program of Jiangsu Science Foundation (BK20232003) and the Fundamental Research Funds for the Central Universities (022114380023).

Contribution Statement

Wen-Da Wei and Xiao-Wen Yang contributed equally to this work. Lan-Zhe Guo supervised the research and is the corresponding author.

References

[Badreddine *et al.*, 2022] Samy Badreddine, Artur S. d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, page 103649, 2022.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, 2009.

[Besold *et al.*, 2021] Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 1–51. 2021.

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

[Dai *et al.*, 2019] Wang-Zhou Dai, Qiu-Ling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. In *Advances in Neural Information Processing Systems*, pages 2811–2822, 2019.

[Fischer *et al.*, 2019] Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin T. Vechev. DL2: training and querying neural networks with logic. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1931–1941, 2019.

[He *et al.*, 2024] Hao-Yuan He, Hui Sun, Zheng Xie, and Ming Li. Ambiguity-aware abductive learning. In *Proceedings of the 41th International Conference on Machine Learning*, 2024.

[Huang *et al.*, 2020] Yu-Xuan Huang, Wang-Zhou Dai, Jian Yang, Le-Wen Cai, Shaofen Cheng, Ruizhang Huang, Yufeng 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*, pages 1070–1075, 2020.

[Huang *et al.*, 2021a] Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. In *Advances in Neural Information Processing Systems*, pages 25134–25145, 2021.

[Huang *et al.*, 2021b] Yu-Xuan Huang, Wang-Zhou Dai, Le-Wen Cai, Stephen H. Muggleton, and Yuan Jiang. Fast abductive learning by similarity-based consistency optimization. In *Advances in Neural Information Processing Systems*, pages 26574–26584, 2021.

[Ionescu *et al.*, 2016] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P. Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2157–2166, 2016.

[Jia *et al.*, 2025] Lin-Han Jia, Wen-Chao Hu, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Verification learning: Make unsupervised neuro-symbolic system feasible. *CoRR*, 2025.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, pages 436–444, 2015.

[Li *et al.*, 2020] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5884–5894, 2020.

[Li *et al.*, 2023] Zenan Li, Zehua Liu, Yuan Yao, Jingwei Xu, Taolue Chen, Xiaoxing Ma, and Jian Lü. Learning with logical constraints but without shortcut satisfaction. In *Proceedings of the The Eleventh International Conference on Learning Representations*, 2023.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, pages 3753–3763, 2018.

[Marconato *et al.*, 2023a] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro-symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. In *Proceedings of the International Conference on Machine Learning*, pages 23915–23936, 2023.

[Marconato *et al.*, 2023b] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In *Advances in Neural Information Processing Systems*, 2023.

[Marconato *et al.*, 2024] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. BEARS make neuro-symbolic models aware of their reasoning shortcuts. *CoRR*, 2024.

[Müller *et al.*, 2019] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019.

[Raedt *et al.*, 2020] Luc De Raedt, Sebastijan Dumancic, Robin Manhaeve, and Giuseppe Marra. From statistical relational to neuro-symbolic artificial intelligence. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 4943–4950, 2020.

[Sangineto *et al.*, 2019] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. Self paced deep learning for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 712–725, 2019.

[Shao *et al.*, 2025] Jie-Jing Shao, Hao-Ran Hao, Xiao-Wen Yang, and Yu-Feng Li. Abductive learning for neuro-symbolic grounded imitation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1221–1232, 2025.

[Soviany *et al.*, 2022] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, pages 1526–1565, 2022.

[Wang *et al.*, 2019] Wei Wang, Isaac Caswell, and Ciprian Chelba. Dynamically composing domain-data selection with clean-data selection by ”co-curricular learning” for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1282–1292, 2019.

[Wang *et al.*, 2023] Kaifu Wang, Efthymia Tsamoura, and Dan Roth. On learning latent models with multi-instance weak supervision. In *Advances in Neural Information Processing Systems*, 2023.

[Xu *et al.*, 2018] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5498–5507, 2018.

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

[Yang *et al.*, 2024] Xiaowen Yang, Wenda Wei, Jie-Jing Shao, Yufeng Li, and Zhi-Hua Zhou. Analysis for abductive learning and neural-symbolic reasoning shortcuts. In *Proceedings of the International Conference on Machine Learning*, 2024.

[Zhou, 2019] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, pages 76101:1–76101:3, 2019.