A Learnability Analysis on Neuro-Symbolic Learning

Hao-Yuan He, **Ming Li**

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{hehy,lim}@lamda.nju.edu.cn

Abstract

This paper analyzes the learnability of neuro-symbolic (NeSy) tasks within hybrid systems. We show that the learnability of NeSy tasks can be characterized by their derived constraint satisfaction problems (DCSPs). Specifically, a task is *learnable* if the corresponding DCSP has a unique solution; otherwise, it is *unlearnable*. For learnable tasks, we establish error bounds by exploiting the clustering property of the hypothesis space. Additionally, we analyze the asymptotic error for general NeSy tasks, showing that the expected error scales with the disagreement among solutions. Our results offer a principled approach to determining learnability and provide insights into the design of new algorithms.

1. Introduction

Neuro-symbolic learning (NeSy) aims to integrate data-driven learning with knowledge-driven reasoning into a unified framework (Hitzler & Sarker, 2022; Marra et al., 2024). Current state-of-the-art NeSy methods primarily adopt a hybrid approach, combining learning and reasoning systems, such as ABL (Zhou, 2019; Dai et al., 2019) and DeepProbLog (Manhaeve et al., 2018; 2021a).

An illustrative prototype is shown in figure 1. Initially, the system employs a learning model, to map input queries $\boldsymbol{x}$ to corresponding concepts $\hat{\boldsymbol{z}}$. Then, a symbolic system (KB), such as a first-order logic solver, processes $\hat{\boldsymbol{z}}$ to deduce the final answer $\hat{y}$. The system evaluates the predicted concepts $\hat{\boldsymbol{z}}$, for instance, by verifying whether $\hat{\boldsymbol{z}} \wedge \texttt{KB} \models y$. Feedback from KB is provided to the learning model in various forms, such as pseudo-labels in ABL or weighted model counting in DeepProbLog, to facilitate further improvements. This prototype can be widely applied in various domains, such as puzzle solving, code generation, and self-driving path planning (Jiao et al., 2024; Li et al., 2024; Hu et al., 2025).

The hybrid system (cf. figure 1) enables end-to-end training in a weakly supervised manner using only $(\boldsymbol{x}, y)$ pairs, i.e., training the model $f : \mathcal{X} \to \mathcal{Z}$ relying solely on the raw data and the final answers. Since the concepts $\boldsymbol{z}$ are not given, NeSy methods aim to minimize the discrepancy between the model's prediction and the knowledge base:

$$R_{\text{NeSy}}(f) = \mathbb{E}_{(\boldsymbol{x},y)} \left[\mathbb{I} \left(f(\boldsymbol{x}) \wedge \texttt{KB} \not\models y \right) \right]. \tag{1}$$

Note that the goal is to learn the model f to generalize well to unseen data, which requires a low concept error. The concept risk is defined using the labeling function g:

$$R_{0/1}(f) = \mathbb{E}_{x} \left[\mathbb{I} \left(f(x) \neq g(x) \right) \right]. \tag{2}$$

In this context, we are concerned with the question of *learnability*. That is, when can (2) be minimized based on the empirical risk minimization over (1) under a finite sample set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, as N increases to infinity? In the learnable case, the NeSy model can learn from a finite-sized dataset D and minimize (2). However, in the unlearnable case, the NeSy model cannot learn from the dataset D even as the sample size approaches infinity.

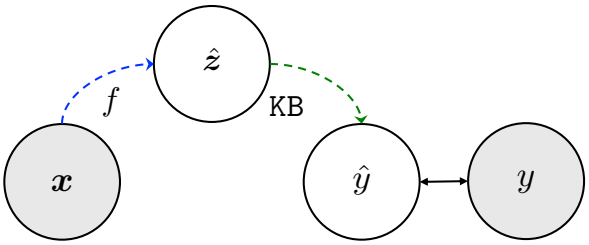


Figure 1: A typical inference process of *hybrid neuro-symbolic system*. Shadowed circles denote observed variables, $\boldsymbol{x}$ is raw input data, $\hat{\boldsymbol{z}}$ is intermediate concepts, $\hat{y}$ is the final answer inferred by KB, and y denotes the true final answer. The goal is to learn the model f.

Consider an unlearnable example: $\texttt{XOR}(\blacksquare, \blacksquare) = 1$, where the background knowledge specifies a logical exclusive-or digit equation. Two hypotheses, $(\blacksquare \mapsto 0, \blacksquare \mapsto 1)$ and $(\blacksquare \mapsto 1, \blacksquare \mapsto 0)$, both satisfy the background knowledge. As a result, the learning model may fail to recognize the correct intermediate concept, as either hypothesis can satisfy the background knowledge. For unlearnable tasks, even an infinite size of training data does not guarantee a good model that minimizes (2). Therefore, *it is crucial to characterize which types of NeSy tasks are (un)learnable.*

In this paper, we characterize the learnability of NeSy tasks within the probably approximately correct (PAC) framework (Valiant, 1984). We find that the key to understanding the learnability of the NeSy task is to construct the NeSy task to a derived constraint satisfaction problem (DCSP). The main result of this paper, the learnability of NeSy tasks, is stated in the following theorem.

Theorem 1.1. *For a neuro-symbolic task $\mathcal{T}$ with a restricted hypothesis space $\mathcal{F}^*$ exhibiting the clustering property, learnability is determined by the conditions:*

- *If the derived constraint satisfaction problem has a unique solution, the task is* learnable. *Specifically, the concept error is bounded by ϵ, given that the sample size N satisfies the following condition:*

$$N > \frac{1}{\kappa} \cdot \log\left(|\mathcal{B}|/\epsilon\right) ,$$

  where $|\mathcal{B}|$ denotes a task-specific constant, and κ is a positive constant determined by the data distribution.

- *Otherwise, the task is* unlearnable.

With the above theorem, by formulating the NeSy task as a constraint satisfaction problem, we can benefit from the advances of the modern solver of constraint satisfaction problems (Prud'homme & Fages, 2022), and easily check whether the task is (un)learnable. Moreover, after obtaining the solution space of the DCSP, we can calculate the disagreement d among the solutions (cf. section 4.3). For general NeSy tasks, in the asymptotic regime of infinitely large sample sizes, the expected error from empirical risk minimization is bounded by d/L, where L represents the size of the concept space.

The remainder of this paper is organized as follows:

In section 3, we review neuro-symbolic learning methods and background. Next, in section 4, we introduce the learnability analysis, covering formal definitions, conditions for learnability, and asymptotic error analysis. We also discuss how previously unlearnable tasks can potentially become learnable under the DCSP framework. Furthermore, in section 5, we validate our theoretical findings through experimental results. Finally, section 6 and section 7 discuss future directions and conclude the paper.

2. Related Works

The combination of learning and reasoning remains the holy grail problem of AI for decades now (Towell & Shavlik, 1994; Sun, 1994; Garcez et al., 2002). One promising approach is to directly incorporate logical constraints into the loss function as the optimization objective (Xu et al., 2018; Roychowdhury et al., 2021; He et al., 2024a). However, since the logical constraint is discrete, the optimization must be projected into the continuous space. This requires an approximation of logical reasoning. Such an approach can lead to issues when approximating discrete logical computations (van Krieken et al., 2022). A more effective approach is the hybrid system, where both the learning and reasoning models function at their

full capacity. For instance, DeepProbLog (Manhaeve et al., 2018; 2021a), NeurASP (Yang et al., 2020) and Scallop (Li et al., 2023a) employ probabilistic logic programming as their reasoning model. ABL (Zhou, 2019; Dai et al., 2019) employs abductive reasoning for logical inference. Recently, there have been some studies on NeSy with auto-regressive or temporal models (Manginas et al., 2025; Smet et al., 2025). Building on the hybrid approach, there have been many successful applications (Mao et al., 2019; Wang et al., 2021; Cai et al., 2021; Verreet et al., 2023; Gao et al., 2024; Jiao et al., 2024). Since the hybrid approach has shown its superiority, it is worthwhile to establish a theoretical framework for analyzing its learnability.

Recently, the "reasoning shortcut" (RSs) (Marconato et al., 2023b;a; Li et al., 2023b; He et al., 2024a; Marconato et al., 2024; Bortolotti et al., 2024) has been observed, referring to the mismatch phenomenon between NeSy risk and concept risk. The underlying reason for this issue is that the specific NeSy task is inherently unlearnable. The analysis of learnability in this paper may help in understanding this phenomenon and for effective algorithm design.

There are also several studies that aim to provide theoretical insights into NeSy methods. For instance, Wang et al. (2023) propose a weakly supervised learning framework, termed multi-instance partial label learning (MI-PLL), to study NeSy learning. However, their approach relies on the condition that, for any $z \in \mathcal{Z}$, there exists a unique y such that $\sigma(z, z, \dots) = y$, which assumes $[z, z, \dots]$ must constitute a valid input to the symbolic system, which often does not hold in practice. Besides, Tao et al. (2024) examine scenarios where randomly selecting abduction candidates results in a consistent optimization objective within the ABL framework. Additionally, Yang et al. (2024) introduce a shortcut risk metric to quantify the gap between true risk and surrogate risk. They establish an upper bound on shortcut risk based on the complexity of the knowledge base. However, their findings do not offer a comprehensive framework for evaluating the learnability of NeSy tasks. Despite prior efforts, a substantial research gap persists in understanding the learnability of NeSy tasks.

3. Preliminaries

Let lowercase letters, e.g., x, z, to denote instances, uppercase letters, e.g., X, Z, to denote random variables, and boldface letters, e.g., $\boldsymbol{x}, \boldsymbol{z}$, to denote vectors or sequences. Let $P(\cdot)$ represents a distribution, while $p(\cdot)$ or $\Pr[\cdot]$ denotes probability, and $[m]$ stands for the set $\{1, \dots, m\}$.

In this section, we first set up the problem. Then we provide a brief review of the principal methods in this field.

3.1. Problem Setup

A typical hybrid neuro-symbolic system consists of two parts: a *machine learning model*, e.g., neural networks, and a *reasoning model*. The learning model $f : \mathcal{X} \to \mathcal{Z}$ maps an instance x (e.g., image, text, or audio) from the input space $\mathcal{X}$ to an intermediate concept z (e.g., primitive facts or predicates) in the symbol space $\mathcal{Z}$, where $|\mathcal{Z}| = L$. The reasoning model $\texttt{KB}$ consists of rules over the concept space and can be implemented using any logic-based system, such as ProbLog (De Raedt et al., 2007) or answer set programming (ASP, Dimopoulos et al. 1997). Assume that a labeling function $g : \mathcal{X} \to \mathcal{Z}$ exists such that $z = g(x)$. The learning model is parameterized by θ, and $p_\theta(\cdot)$ represents the likelihood estimated by the model, where $f(x) = \arg\max_{z \in \mathcal{Z}} p_\theta(z \mid x)$.

To the inference process (cf. figure 1), the learning model f accepts multiple instances as a sequence $\boldsymbol{x} = (x_1, \dots, x_m)$ and outputs a sequence of concepts $\hat{\boldsymbol{z}} = (\hat{z}_1, \dots, \hat{z}_m)$, where m denotes the number of instances. The output $\hat{\boldsymbol{z}}$ is then passed to the reasoning model $\texttt{KB}$, which infers the final label $y \in \mathcal{Y}$ through logical entailment, i.e., $\hat{\boldsymbol{z}} \wedge \texttt{KB} \models y$. To simplify the inference process of $\texttt{KB}$, we represent it by a logical forward operator $\sigma(\cdot)$ such that $\sigma(\hat{\boldsymbol{z}}) = y$. In a standard neuro-symbolic learning setup (Dai et al., 2019; Manhaeve et al., 2021a; Li et al., 2023a; Marra et al., 2024), the training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ are sampled from data distribution $\mathcal{D} = (\mathcal{X}^m, \mathcal{Y})$. Therefore, a neuro-symbolic task can be formally defined as a triple $\mathcal{T} = \langle \mathcal{X}, \mathcal{Y}, \texttt{KB} \rangle$.

Example 1 (Addition)**.** The input $\boldsymbol{x} \in \mathcal{X}^2$, where $\mathcal{X}$ denotes digit images. The concepts $\mathcal{Z}$ consist of digits from `0` to `9`. The data takes the form ( ,) $\mapsto$ `1`, with the logical forward function $\sigma(\cdot) := \texttt{SUM}(\cdot, \cdot)$. The label space $\mathcal{Y}$ is defined as the addition results, i.e., from `0` to `18`.

Notably, when all final labels y are identical (e.g., $\boldsymbol{z} \wedge \texttt{KB} \models \top$), such as in the case of code generation where all code must satisfy a syntax constraint (Jiao et al., 2024), the final label y can be omitted for simplicity. The analysis presented in this paper can be readily extended to such cases.

The goal of NeSy is to identify the optimal learning model for concept recognition, making it essential to evaluate its performance at the concept level.

Definition 3.1. The concept risk is defined as follows:

$$R_{0/1}(f;g) = \mathbb{E}_x \left[\mathbb{I}\left(f(x) \neq g(x)\right)\right]. \tag{3}$$

For simplicity, we omit g and denote (3) as $R_{0/1}(f)$. However, optimizing (3) is challenging due to the lack of supervision regarding the intermediate concept.

3.2. Neuro-Symbolic Methods

To minimize (3), the key idea of current state-of-the-arts methods (Manhaeve et al., 2018; Dai et al., 2019) is to optimize the neuro-symbolic risk as surrogate, which aims to minimize the discrepancy between the learning and reasoning models.

Definition 3.2. The neuro-symbolic risk is formally defined as follows:

$$R_{\text{NeSy}}(f) = \mathbb{E}_{(\boldsymbol{x},y)} \left[\mathbb{I}\left(f(\boldsymbol{x}) \wedge \texttt{KB} \not\models y\right)\right]. \tag{4}$$

The learning process of NeSy is to select the optimal function $f^* \in \mathcal{F}$ that minimizes the NeSy risk; that is:

$$f^* = \arg\min_{f \in \mathcal{F}} R_{\text{NeSy}}(f). \tag{5}$$

Probabilistic Neuro-Symbolic Learning. Probabilistic neuro-symbolic learning (PNL, Manhaeve et al. 2021b) methods adopt reasoning models via probabilistic logic programming, such as DeepProblog (Manhaeve et al., 2018; 2021a), NeurASP (Yang et al., 2020), and Scallop (Li et al., 2023a). Because (4) is discrete and challenging to optimize, PNL aims to minimize the following objective:

$$-\mathbb{E}_{(\boldsymbol{x},y)} \Pr\left[y \mid \boldsymbol{x}; f, \texttt{KB}\right]. \tag{6}$$

Reformulating (6), we can express the objective as follows:

$$-\mathbb{E}_{(\boldsymbol{x},y)} \log \sum_{\boldsymbol{z}} \mathbb{I}(\boldsymbol{z} \wedge \texttt{KB} \models y) \cdot \Pr\left[\boldsymbol{z} \mid \boldsymbol{x}; f, \texttt{KB}\right]. \tag{7}$$

The above objective is referred to as the probabilistic neuro-symbolic learning risk, denoted as $R_{\text{PNL}}(f)$.

The key operation for calculating the PNL risk is $\sum_{\boldsymbol{z}} \mathbb{I}(\boldsymbol{z} \wedge \texttt{KB} \models y) \cdot \Pr\left[\boldsymbol{z} \mid \boldsymbol{x}; f, \texttt{KB}\right]$, also well-known as *weighted model counting* (WMC), which requires enumerating all possible worlds that satisfy the constraints of the symbolic system. This operation can be performed using various approaches, such as ProbLog (De Raedt et al., 2007) or answer set programming (Dimopoulos et al., 1997). However, in general, the computational complexity of WMC is #P (Maene et al., 2024), which makes PNL methods challenging to scale.

Abductive Learning. Unlike PNL methods, abductive learning methods (Dai et al., 2019; Huang et al., 2021; Hu et al., 2025) infer the *most* plausible concepts through abductive reasoning and use them to update the model. The objective of ABL is to minimize the ABL risk:

$$R_{\text{ABL}}(f) = -\mathbb{E}_{(\boldsymbol{x},y)} \log\left(\Pr\left[y, \bar{\boldsymbol{z}} \mid \boldsymbol{x}; f, \texttt{KB}\right]\right), \tag{8}$$

where $\bar{\boldsymbol{z}} = \min_{\boldsymbol{z} \in A(y)} \text{Score}(\boldsymbol{z}, f(\boldsymbol{x}))$ represents the most likely candidate in the abduction set. The abduction set $A(y)$ includes all possible concepts $\boldsymbol{z}$ that satisfy the constraints of KB and have a non-zero measure, i.e., $\Pr[\boldsymbol{z}] > 0$. The score function measures the alignment between a candidate $\boldsymbol{z}$ and the model's prediction $f(\boldsymbol{x})$. For instance, Dai et al. (2019) propose using the Hamming distance (Hamming, 1950) as the score function.

ABL enhances computational efficiency by concentrating on the most plausible candidates, thereby avoiding the enumeration of all possible worlds. However, the inherent ambiguity in the abduction process can lead to incorrect candidate selection (Magnani, 2009), introducing bias into the learning objective.

Unified View. Both PNL and ABL can effectively optimize (4). Formally, we can state the following theorem:

Theorem 3.3. *A minimizer of R_{PNL} or R_{ABL} is also a minimizer of R_{NeSy}. For each surrogate $R_s \in \{R_{PNL}, R_{ABL}\}$, we have:*

$$\arg\min_{f\in\mathcal{F}} R_s(f) \subseteq \arg\min_{f\in\mathcal{F}} R_{NeSy}(f)\,.$$

The proof of this theorem is presented in appendix A.1.

4. Learnability of Neuro-Symbolic Learning

Recall that the learnability we discussed is, when can the concept risk (3) be minimized based on a finite sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, with empirical NeSy risk (4) minimization as $N \to \infty$? While sometimes this is possible, other times it is not. To understand why this happens, it is crucial to establish a learnability analysis: *what kinds of NeSy task is learnable?* Similar to standard probably approximately correct (PAC) learning (Valiant, 1984), the learnability of a NeSy task is defined as follows:

Definition 4.1. Let N denote the size of samples drawn i.i.d. from $\mathcal{D}$, $\mathcal{T}$ represent a NeSy task, and $\mathcal{F}$ denote the hypothesis space. We say that $\mathcal{T}$ is *learnable* if: for any $0 < \epsilon,\ \delta < 1$ and distribution $\mathcal{D}$, there exists an algorithm $\mathcal{A}$ and an integer $N_{\epsilon,\delta}$ such that, whenever $N \geq N_{\epsilon,\delta}$, the selected hypothesis $\hat{f}$ satisfies $\Pr[R_{0/1}(\hat{f}) \leq \epsilon] \geq 1-\delta$. Otherwise, we say that it is *unlearnable*.

We focus on the empirical risk minimization (ERM) algorithm during the analysis, since it is found that in the common learning setting, e.g., supervised classification and regression, a problem is learnable if and only if it is learnable by ERM (Blumer et al., 1989; Alon et al., 1997).

4.1. Restricted Hypothesis Space

Unlike supervised learning, where the goal is to learn a hypothesis mapping $x \mapsto y$ from the (x, y) pair, the NeSy task aims to learn a mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$ from the $(\boldsymbol{x}, y)$ pair. Ideally, each y is expected to correspond to a unique $\boldsymbol{z}$, ensuring that $\forall y \in \mathcal{Y}, |A(y)| = 1$. Under this condition, the task reduces to a standard learning problem (Vapnik, 1999). However, this condition is rarely held in practice, making NeSy tasks inherently difficult to learn. When multiple plausible solutions $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ exist, the learning task becomes more complex owing to inherent ambiguity; we define the task as *ambiguous* if there exists some $y \in \mathcal{Y}$ such that $|A(y)| \geq 2$.

Proposition 4.2. *For an ambiguous NeSy task $\mathcal{T}$, if the hypothesis space $\mathcal{F}$ is sufficiently complex (e.g., capable of shattering the task), there exists f^* that minimizes R_{NeSy} but does not minimize the $R_{0/1}$.*

The proof is provided in appendix A.2. Proposition 4.2 suggests that ambiguous NeSy tasks may be unlearnable when the hypothesis space is very complex, such as nearest neighbor, whose Vapnik-Chervonenkis dimension is infinite (Karacali & Krim, 2003), or deep neural networks (Bartlett & Maass, 2003) without any regularization terms. This scenario arises due to overfitting caused by the high memorization capacity of models (Zhang et al., 2021). To mitigate this problem, it is necessary to constrain the complexity of the hypothesis space. One approach is to leverage self-supervised learning methods, which promote clustering properties in neural networks, thereby improving generalization (Sohn et al., 2020).

Consider a scenario where a pre-trained model satisfies a clustering property, meaning that instances representing the same concept are grouped together in feature space. In the ambiguous task described in example 1, if the model correctly processes a key sample such as SUM( , ) = 0, it can reliably identify 0. This, in turn, simplifies subsequent tasks. For example, once the model recognizes SUM( ,) = 1, it can correctly identify 1. By iteratively applying this process, the model can learn to recognize all relevant concepts despite initial ambiguity.

The above process highlights the need to restrict the hypothesis space for the learning system. This hypothesis space ensures consistent mappings between concepts and labels, which can be formalized as follows:

Definition 4.3. Let $\mathcal{F}^*$ be a restricted hypothesis space which ensures that instances with the same label correspond to the same concept, and vice versa. Given the labeling function g, formally, for any $f \in \mathcal{F}^*$:

$$\forall x_1, x_2 \in \mathcal{X}, \quad g(x_1) = g(x_2) \iff f(x_1) = f(x_2).$$

4.2. Derived Constraint Satisfaction Problem

The restricted hypothesis space implicitly partitions the raw input space $\mathcal{X}$ into L clusters. Here we use $\langle x \rangle_i$ to denote the cluster $\{x \mid x \in \mathcal{X}, f(x) = i\}$. The learning process is to establish a mapping between the clusters $\{\langle x \rangle_1, \ldots, \langle x \rangle_L\}$ and $\mathcal{Z}$ that minimize the NeSy risk. This process inherently transforms the NeSy learning problem into a *constraint satisfaction problem* (CSP). In this paper, we referred to it as derived CSP (DCSP).

Definition 4.4. The derived constraint satisfaction problem for a NeSy task $\mathcal{T}$ is defined as a triple $\langle \mathsf{V}, \mathsf{D}, \mathsf{C} \rangle$, where:

- $\mathsf{V} = \{V_1, \ldots, V_L\}$ are the variables,
- $\mathsf{D} = \{D_1 = \mathcal{Z}, \ldots, D_L = \mathcal{Z}\}$ are the domains, and
- $\mathsf{C} = \{C_1, \ldots, C_N\}$ are the constraints.

Each V_i corresponds to a mapping from $\langle x \rangle_i$ to a concept label. For convenience, we slightly abuse notation by letting $\mathsf{V}(\boldsymbol{x})$ denote a mapping from an input sequence to the corresponding concept sequence determined by the mapping set V. Each C_j corresponds to a constraint $(\boldsymbol{x}_j, y_j)$, e.g., $\mathsf{V}(\boldsymbol{x}_j) \wedge \mathtt{KB} \models y_j$. Solving the DCSP is to find a consistent assignment I that satisfies all constraints.

A DCSP solution I corresponds to an assignment of values to variables, expressed as $I = \{(V_1, v_1), \ldots, (V_L, v_L)\}$, where each v_i is the value assigned to the variable V_i. For simplicity, we denote the solution as $I = (v_1, \ldots, v_L)$ by omitting the variables. Here we only discuss the case when the DCSP has solution(s); Otherwise, the learning model will inevitably conflict with the background $\mathtt{KB}$.

4.3. Conditions of Learnability

In general, the solution to a DCSP may not be unique, meaning multiple distinct solutions can exist. We represent it as a solution space $\mathcal{S} = \{I_1, \ldots, I_k\}$. We use Union() to represent the common assignments among the solutions. When the given input set has only one element, this operation simply returns this element.

Definition 4.5. The *DCSP solution disagreement* d quantifies the inconsistency among all solutions:

$$d = L - |\text{Union}(\mathcal{S})| \,.$$

The disagreement d measures how many variables have different values across the solutions in $\mathcal{S}$. If $d = 0$, which means $|\mathcal{S}| = 1$, there is only one solution. This means the optimal hypothesis can be determined by minimizing the NeSy risk, aligning with the concept risk.

Lemma 4.6. *For a NeSy task $\mathcal{T}$, if the DCSP solution disagreement $d = 0$, then the NeSy risk is equivalent to the concept risk. Formally, for any $f \in \mathcal{F}$:*

$$R_{\textit{NeSy}}(f) \to 0 \iff R_{\textit{0/1}}(f) \to 0.$$

Proof Sketch. The direction from the right-hand side to the left-hand side is straightforward; here, we focus on proving the reverse direction. We demonstrate this by showing that *if the concept risk is non-zero, then the NeSy risk cannot be zero* (contraposition). If the concept risk is non-zero, there must be at least one misclassified instance where f assigns an incorrect label. Given that the DCSP solution is unique and there is no disagreement (i.e., $d = 0$), any such misclassification directly results in a non-zero NeSy risk. Therefore, if the NeSy risk is zero, it follows that the concept risk must also be zero. □

The detailed proof is provided in appendix A.3. To further investigate the learnability of NeSy tasks, we introduce the following mild assumptions.

Assumption 4.7. (Finite Cardinality) *The set of possible concept sequences, $\mathcal{B} = \bigcup_{y \in \mathcal{Y}} A(y)$, has finite cardinality.*

Assumption 4.8. (Non-vanishing Probability) *The sampling process is controlled by a distribution $P(Z)$, and for any concept sequence $\boldsymbol{z} \in \mathcal{B}$, the probability of being sampled is at least a small positive constant κ.*

Under these assumptions, we restate the main theorem 1.1, which was informally introduced in section 1.

Theorem 4.9. *For a neuro-symbolic task $\mathcal{T}$ with a restricted hypothesis space $\mathcal{F}^*$ exhibiting the clustering property, learnability is determined by the conditions:*

- *If the derived constraint satisfaction problem has a unique solution, the task is* learnable*. Specifically, the concept error is bounded by ϵ, given that the sample size N satisfies the following condition:*

$$N > \frac{1}{\kappa} \cdot \log\left(|\mathcal{B}|/\epsilon\right) .$$

- *Otherwise, the task is* unlearnable*.*

The proof is in appendix A.4 . Theorem 4.9 establish that a NeSy task $\mathcal{T}$ is *learnable* if and only if the DCSP solution is unique, i.e., disagreement $d = 0$. Conversely, if the DCSP has multiple solutions (i.e., $d \geq 1$), the task is *unlearnable*, implying that concept error remains *unavoidable* regardless of additional training data.

Building upon the concept of DCSP solution disagreement, we derive a more general theorem offering deeper insights into learning errors in a restricted hypothesis space $\mathcal{F}^*$. Consider a learning process that minimizes the empirical NeSy risk $\hat{R}_{\text{NeSy}}$. As the sample size approaches infinity, the hypotheses learned via empirical risk minimization (ERM) asymptotically converge to:

$$\mathcal{F}^*_{\text{ERM}} = \arg\min_{f \in \mathcal{F}^*} R_{\text{NeSy}}(f).$$

The average error of the ERM result, denoted by $\mathcal{E}^*$, is the expected concept risk of the arbitrarily selected hypothesis:

$$\mathcal{E}^* = \mathbb{E}_{f \in \mathcal{F}^*_{\text{ERM}}} \left[R_{0/1}(f)\right] .$$

Theorem 4.10. *The average error $\mathcal{E}^*$ is bounded by:*

$$\mathcal{E}^* \leq \frac{d}{L} .$$

Proof. Recall that the DCSP solution disagreement d is given by $d = L - \text{Union}(\mathcal{S})$, where $\text{Union}(\mathcal{S})$ represents the common assignments among the solutions. Since the restricted hypothesis space ensures that instances with the same assigned label correspond to the same concept and vice versa. In the worst case, errors occur in at d classes, so the maximum true risk is $\max_{f \in \mathcal{F}^*_{\text{ERM}}} R_{0/1}(f) = d/L$. Thus, the average error is bounded by: $\mathcal{E}^* \leq \max_{f \in \mathcal{F}^*_{\text{ERM}}} R_{0/1}(f) = d/L$. □

Theorem 4.10 provides an asymptotic error analysis for NeSy tasks, indicating that as the DCSP solution disagreement d increases, the upper bound of the concept error also increases. Revealing that the disagreement d is the key to understanding the learnability of NeSy tasks.

4.4. Examples

Here we present some examples for better understanding the learnability conditions of a NeSy task. To demonstrate the distinction between *learnable* and *unlearnable* tasks, we utilize digital images as input data to give a few examples. The data is modeled as $\boldsymbol{x} = (x_1, x_2) \in \mathcal{X}^2$, where $\mathcal{X}$ represents the space of digit images, e.g., $\{$  ,  , $\dots\}$. The intermediate concept space $\mathcal{Z}$ and the label space $\mathcal{Y}$ depend on the specific knowledge base.

Table 1 provides a summary of the examples. Learnable tasks are straightforward to verify, as prior research has demonstrated the effectiveness of NeSy methods for these tasks (Manhaeve et al., 2021a; Li et al., 2023a; He et al., 2024b). For the unlearnable cases, we present two distinct solutions to the DCSP for each example.

For the XOR task ($d/L = 1$), if the concepts `0` and `1` are interchanged, e.g., ($\mapsto$ `0`, $\mapsto$ `1`) and ($\mapsto$ `1`, $\mapsto$ `0`), the NeSy risk can be minimized.

For the modular addition task ($k = 9, d/L = 0.2$), if the mappings of `0` and `9` are swapped, e.g., ($\mapsto$ `0`, $\mapsto$ `9`) and ($\mapsto$ `9`, $\mapsto$ `0`), the NeSy risk can be minimized.

4.5. Ensemble Unlearnable Tasks

Some NeSy tasks are inherently unlearnable because they admit multiple solutions to their DCSPs, resulting in ambiguity. This ambiguity cannot be resolved by increasing data or improving the learning algorithm, as it stems from intrinsic task properties. Interestingly, such unlearnable tasks may become learnable when combined in an ensemble framework under a multi-task learning paradigm. The key insight is that tasks can mutually constrain each other, reducing ambiguity.

Table 1: Examples of learnable and unlearnable tasks. In specific, the modular addition task requires $2 \leq k \leq 10$.

Category	Task	Knowledge Base
Learnable	Addition	$y = z_1 + z_2$
	Multiplication	$y = z_1 \times z_2$
Unlearnable	Exclusive OR	$y = z_1 \oplus z_2$
	Modular Addition	$y = (z_1 + z_2) \texttt{ mod } k$

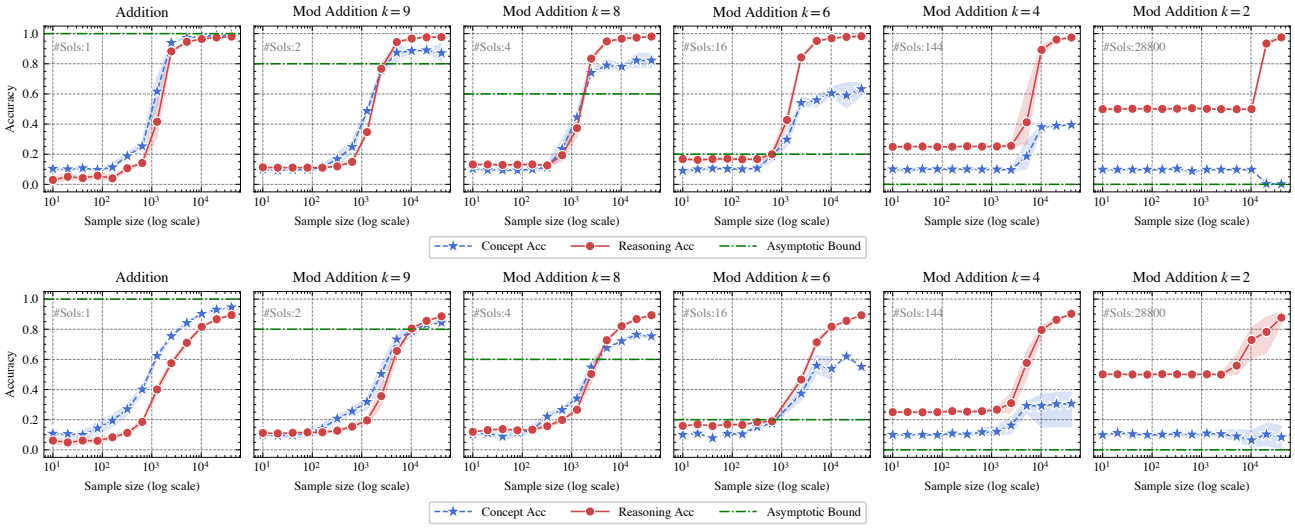


Figure 2: *Accuracies* versus *sample size* for different NeSy tasks (top MNIST and bottom KMNIST). The shadowed area denotes the standard error. The number of the DCSP solutions (#Sols) is shown at the top left of each plot. The asymptotic bound (green line) from theorem 4.10 indicates that concept accuracy should exceed this bound as the sample size grows.

Consider two unlearnable tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ with corresponding solution spaces $\mathcal{S}_1$ and $\mathcal{S}_2$. Each task is individually unlearnable, i.e., $|\mathcal{S}_1| \geq 2$ and $|\mathcal{S}_2| \geq 2$. In a multi-task learning setting, the model is required to satisfy both tasks simultaneously, thereby constraining the solution to the intersection $\mathcal{S}_1 \cap \mathcal{S}_2$. This intersection reduces ambiguity and can potentially yield a unique solution. Therefore, by leveraging the unique constraints induced by the intersection of solution spaces, combining unlearnable tasks into an ensemble framework may enable learning.

From the perspective of DCSP, we can formally state the conditions when tasks can ensemble to become learnable:

Corollary 4.11. *NeSy tasks become learnable in an ensemble framework if combining their DCSPs results in a unique solution.*

5. Empirical Study

To empirically validate the theoretical results, we conducted a series of experiments. Due to space limitations, some experimental results are presented in the appendix.

Setup Manhaeve et al. (2018) proposed the Addition task by incorporating the handwritten MNIST (Deng, 2012) and predefined addition rules. We extend the setup by including KMNIST (Clanuwat et al., 2018), CIFAR10 (Krizhevsky, 2009), and SVHN (Netzer et al., 2011), mapping class indices to digits and enriching the background knowledge as depicted in table 1. The learning model for MNIST and KMNIST is LeNet (LeCun & Bengio, 1998), while ResNet50 (He et al., 2016) is used for CIFAR10 and SVHN. The results were obtained using an Intel Xeon Platinum 8538 CPU and an NVIDIA A100-PCIE-40GB GPU on an Ubuntu 20.04 platform. All experiments were conducted five times with different random seeds. More details can be seen in appendix B.

Method To effectively optimize the NeSy risk (4), we adopt the following surrogate (cf. proof in appendix A.1):

$$-\mathbb{E}_{(\boldsymbol{x},y)} \log \left(\sum_{\bar{\boldsymbol{z}} \in \mathcal{N}(y)} \Pr\left[y, \bar{\boldsymbol{z}} \mid \boldsymbol{x}; f, \mathtt{KB}\right] \right), \tag{9}$$

which is flexible, where $\mathcal{N}(y) \subseteq A(y)$ represents several valid candidates for the final answer y. By restricting the size of $\mathcal{N}(y)$ from the entire set $A(y)$ to the most likely candidate $\bar{\boldsymbol{z}}$, we achieve a balance between PNL and ABL, and we set the size of $\mathcal{N}(y)$ is $\min(16, |A(y)|)$. The implementation is based on the code of He et al. (2024b). For brevity, detailed experiments on PNL and ABL are provided in appendix B.

5.1. Empirical Analysis on Learnability

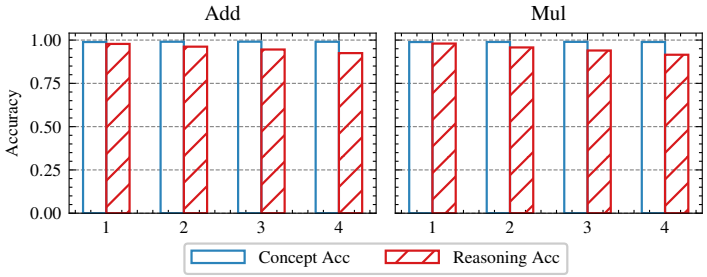


Figure 3: Accuracies on the learnable tasks.

We empirically evaluate the learnability of NeSy tasks based on theorem 4.9, focusing on two key aspects: (i) validating that minimizing the NeSy risk consistently minimizes the concept risk for learnable tasks, and (ii) examining how DCSP solution disagreement affects learnability.

(i) Validation of Learnable Tasks. We first validate the learnability conditions (cf. theorem 4.9) by examining addition and multiplication tasks (cf. table 1). By solving the DCSP shows that both tasks are learnable, and their learnability remains unaffected by increases in digit size (e.g., from $\mathtt{PROD}(\square, \square) = 2$ to $\mathtt{PROD}(\square\square, \square\square) = 200$). The raw dataset in figure 3 is MNIST, and additional results for other datasets are in the appendix. We further substantiate learnability by examining tasks with varying digit sizes, ranging from one to four digits. As depicted in figure 3, the results confirm that: (a) Optimization of the surrogate risk (9) effectively minimizes the NeSy risk. (b) For learnable tasks, a good minimizer of the NeSy risk also serves as a reliable minimizer of the concept risk.

(ii) Impact of DCSP Solution Disagreement. We further investigate how disagreement in DCSP solutions impacts learnability. According to theorem 4.10, the asymptotic error is bounded by the ratio of DCSP solution disagreement d to the size of the concept space L. Experiments involving addition and modular addition tasks with varying modular bases k reveal that altering the knowledge base changes the DCSP solution space, directly influencing learnability. For clarity, we plot the asymptotic accuracy bound for each task, i.e., $1 - d/L$, showing that higher disagreement results in a lower bound line (green). As shown in figure 2: (a) Tasks with a unique DCSP solution are learnable; (b) Tasks with high DCSP disagreement struggle to achieve low concept risk, even as the sample size increases.

In summary, our empirical analysis confirms the theoretical learnability conditions by demonstrating that minimizing the NeSy risk reliably minimizes the concept risk for learnable tasks; Furthermore, tasks with lower disagreement exhibit better learnability, while those with high disagreement suffer from ambiguity due to multiple conflicting solutions.

5.2. Ensemble of Unlearnable NeSy Tasks

Certain NeSy tasks are inherently unlearnable because their DCSPs admit multiple solutions, resulting in inherent ambiguity. However, when combined in an ensemble framework within a multi-task learning setting, such tasks may become learnable by enforcing mutual consistency, as shown in corollary 4.11. We evaluate corollary 4.11 using mod addition tasks with mod bases k_1 and k_2 under two specific configurations: an unlearnable ensemble ($k_1 = 2$, $k_2 = 3$) and a learnable ensemble ($k_1 = 3$, $k_2 = 4$). For $k = 2, 3, 4$, the degree of DCSP solution disagreement d is 10. The experiments in figure 4 are based on the raw MNIST dataset. Additional configuration details and experiments are provided in appendix B.2.3.

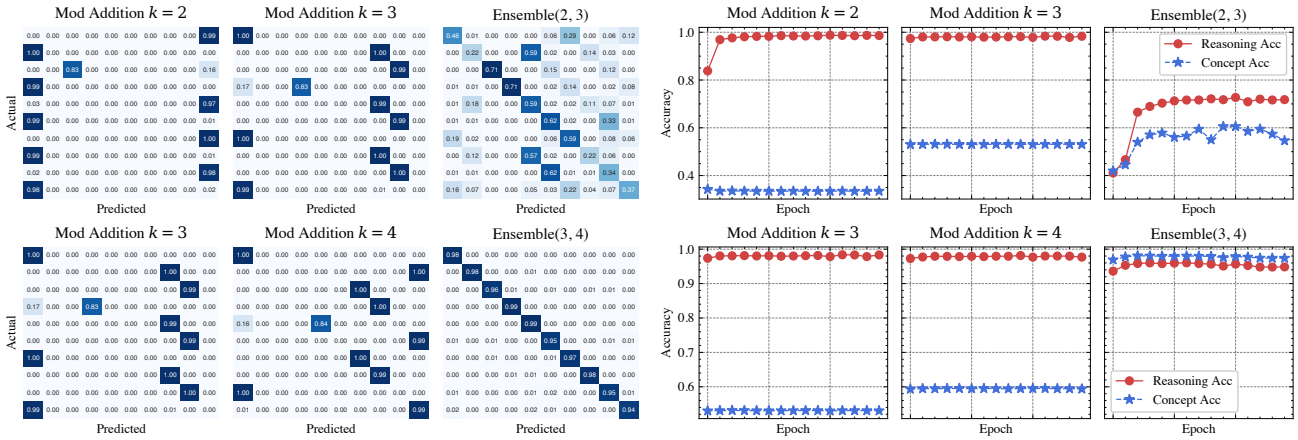


Figure 4: *Ensemble* of unlearnable NeSy tasks. The left shows confusion matrices, and the right displays accuracy curves. (a) The top row illustrates an *unlearnable* case, where combining the tasks still results in multiple DCSP solutions. (b) The bottom row illustrates a *learnable* case, where combining the tasks reduces the DCSP solutions to a single one.

In the top of figure 4, the unlearnable case ($k_1 = 2$, $k_2 = 3$) shows that while the ensemble narrows the solution space, reducing the disagreement d to 8, it does not converge to a unique solution, and the task remains unlearnable.

In the bottom of figure 4, the learnable case ($k_1 = 3$, $k_2 = 4$) illustrates that both tasks initially admit multiple DCSP solutions, causing reasoning accuracy to exceed concept accuracy, as shown in figure 4. Following the ensemble, the intersection of solution spaces yields a unique solution, with the disagreement d reduced to 0, rendering the ensemble task learnable.

This experimental result supports corollary 4.11, demonstrating that forming ensembles of different NeSy tasks can enhance learnability by mutually constraining DCSP solution spaces. This finding suggests that we can collect tremendous NeSy tasks and jointly learn them in an ensemble manner, which could potentially introduce a "scaling law" (Kaplan et al., 2020) in the NeSy domain.

6. Limitation

This paper focuses exclusively on hybrid neuro-symbolic systems, e.g., probabilistic neuro-symbolic and abductive learning methods. Thus the findings may not directly extend to other types of neuro-symbolic methods. The analysis of this study relies on a restricted hypothesis space with a clustering property, which is inherently satisfied by models such as neural networks equipped with manifold regularization (Belkin et al., 2006) or self-supervised pretraining (Liu et al., 2021). However, extending the framework to encompass more general hypothesis spaces without requiring this specific property remains an open challenge.

Future work may involve a deeper investigation into extending the learnability framework to encompass a broader range of NeSy systems. Additionally, exploring the learnability of the semi-supervised case of NeSy tasks, where some training examples are supervised for intermediate concepts, could be an interesting direction. Developing practical strategies for constructing effective task ensembles represents a promising avenue for improving learnability in diverse and complex scenarios.

7. Conclusion

We disclose that a neuro-symbolic (NeSy) task is learnable if and only if the derived constraint satisfaction problem (DCSP) has a unique solution. Using the DCSP framework, we can determine whether a NeSy task is learnable and derive the task-specific asymptotic bound for the concept error. Moreover, we find that constructing ensembles of previously unlearnable tasks reduces the degree of ambiguity, thereby enhancing overall task learnability. Experimental results validate our theoretical findings.

Impact Statement

This paper presents a theoretical analysis of the *learnability* of neuro-symbolic learning, with a particular focus on hybrid systems such as probabilistic neuro-symbolic learning and abductive learning methods. The impact of this work lies in providing theoretical insights into these systems and elucidating the underlying mechanisms of the recently observed "reasoning shortcut" phenomenon. We do not anticipate that this work will introduce any negative ethical or social impacts.

References

Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. Scale-sensitive Dimensions, Uniform convergence, and Learnability. *Journal of the ACM*, 44(4):615–631, July 1997.

Bartlett, P. L. and Maass, W. Vapnik-Chervonenkis Dimension of Neural Nets. *The handbook of brain theory and neural networks*, pp. 1188–1192, 2003.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold Regularization: A geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7(11), 2006.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. Learnability and The Vapnik-Chervonenkis Dimension. *Journal of the ACM*, 36(4):929–965, October 1989.

Bortolotti, S., Marconato, E., Carraro, T., Morettin, P., van Krieken, E., Vergari, A., Teso, S., and Passerini, A. A Neuro-Symbolic Benchmark Suite for Concept Quality and Reasoning Shortcuts. In *Advances in Neural Information Processing Systems 37*, 2024. Datasets and Benchmarks Track.

Cai, L.-W., Dai, W.-Z., Huang, Y.-X., Li, Y.-F., Muggleton, S., and Jiang, Y. Abductive Learning with Ground Knowledge Base. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 1815–1821, Virtual, 2021.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep Learning for Classical Japanese Literature. *CoRR*, abs/1812.01718, 2018.

Dai, W.-Z., Xu, Q., Yu, Y., and Zhou, Z.-H. Bridging Machine Learning and Logical Reasoning By Abductive Learning. In *Advances in Neural Information Processing Systems 32*, pp. 2815–2826, Vancouver, BC, Canada, 2019.

De Raedt, L., Kimmig, A., and Toivonen, H. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2468–2473, Hyderabad, India, 2007.

Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dimopoulos, Y., Nebel, B., and Koehler, J. Encoding Planning Problems in Nonmonotonic Logic Programs. In *Proceedings of the 4th European Conference on Planning*, pp. 169–181, Toulouse, France, 1997.

Gao, E.-H., Huang, Y.-X., Hu, W.-C., Zhu, X.-H., and and, W.-Z. D. Knowledge-enhanced Historical Document Segmentation and Recognition. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, BC, Canada, 2024.

Garcez, A. S. d., Gabbay, D. M., and Broda, K. B. *Neural-Symbolic Learning System: Foundations and Applications*. Springer-Verlag, Berlin, Heidelberg, 2002.

Hamming, R. W. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

He, H.-Y., Dai, W.-Z., and Li, M. Reduced Implication-bias Logic Loss for Neuro-symbolic Learning. *Machine Learning*, 113:3357–3377, 2024a.

He, H.-Y., Sun, H., Xie, Z., and Li, M. Ambiguity-aware Abductive Learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18019–18042, Vienna, Austria, 2024b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

Hitzler, P. and Sarker, M. K. (eds.). *Neuro-Symbolic Artificial Intelligence: The State of the Art*. IOS Press, Amsterdam, 2022.

Hu, W.-C., Dai, W.-Z., Jiang, Y., and Zhou, Z.-H. Efficient Rectification of Neuro-Symbolic Reasoning Inconsistencies by Abductive Reflection. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, Philadelphia, USA, 2025. To appear.

Huang, Y.-X., Dai, W.-Z., Cai, L.-W., Muggleton, S. H., and Jiang, Y. Fast Abductive Learning by Similarity-based Consistency Optimization. In *Advances in Neural Information Processing Systems 34*, pp. 26574–26584, Virtual, 2021.

Huang, Y.-X., Hu, W.-C., Gao, E.-H., and Jiang, Y. ABLkit: A Python Toolkit for Abductive Learning. *Frontiers of Computer Science*, 18(6):186354, 2024.

Jiao, Y., De Raedt, L., and Marra, G. Valid Text-to-SQL Generation with Unification-Based DeepStochLog. In *Neural-Symbolic Learning and Reasoning: 18th International Conference, NeSy 2024, Barcelona, Spain, September 9–12, 2024, Proceedings, Part I*, pp. 312–330, Berlin, Heidelberg, 2024. Springer-Verlag.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Karacali, B. and Krim, H. Fast Minimization of Structural Risk by Nearest Neighbor Rule. *IEEE Transactions on Neural Networks*, 14(1):127–137, 2003.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *The 2nd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. *Technical Report, Department of Computer Science, University of Toronto*, 2009.

LeCun, Y. and Bengio, Y. Convolutional Networks for Images, Speech, and Time Series. In *The Handbook of Brain Theory and Neural Networks*, pp. 255–258. MIT Press, Cambridge, MA, USA, 1998.

Li, Z., Huang, J., and Naik, M. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*, 7(PLDI), June 2023a.

Li, Z., Liu, Z., Yao, Y., Xu, J., Chen, T., Ma, X., and Lyu, J. Learning with Logical Constraints but without Shortcut Satisfaction. In *The 11st International Conference on Learning Representations*, 2023b.

Li, Z., Huang, Y., Li, Z., Yao, Y., Xu, J., Chen, T., Ma, X., and Lü, J. Neuro-symbolic learning yielding logical constraints. In *Advances of the 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2024. Curran Associates Inc.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-Supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

Maene, J., Derkinderen, V., and De Raedt, L. On the Hardness of Probabilistic Neurosymbolic Learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34203–34218, Vienna, Austria, 2024.

Magnani, L. *Abductive Cognition - The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*, volume 3 of *Cognitive Systems Monographs*. Springer, Berlin, Heidelberg, 2009.

Manginas, N., Paliouras, G., and Raedt, L. D. NeSyA: Neurosymbolic Automata. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, Philadelphia, USA, 2025. To appear.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems 31*, pp. 3753–3763, Montréal, Canada, 2018.

Manhaeve, R., Dumančić, S., Kimmig, A., Demeester, T., and De Raedt, L. Neural Probabilistic Logic Programming in DeepProbLog. *Artificial Intelligence*, 298(C):103504, 2021a.

Manhaeve, R., Marra, G., Demeester, T., Dumancic, S., Kimmig, A., and De Raedt, L. Neuro-Symbolic AI = Neural + Logical + Probabilistic AI. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342 of *Frontiers in Artificial Intelligence and Applications*, pp. 173–191. IOS Press, 2021b.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *The 7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019.

Marconato, E., Bontempo, G., Ficarra, E., Calderara, S., Passerini, A., and Teso, S. Neuro-Symbolic Continual Learning: Knowledge, Reasoning Shortcuts and Concept Rehearsal. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 23915–23936. JMLR.org, 2023a.

Marconato, E., Teso, S., Vergari, A., and Passerini, A. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In *Advances in Neural Information Processing Systems*, volume 36, pp. 72507–72539. Curran Associates, Inc., 2023b.

Marconato, E., Bortolotti, S., van Krieken, E., Vergari, A., Passerini, A., and Teso, S. Bears make neuro-symbolic models aware of their reasoning shortcuts. In *Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 2399–2433. PMLR, 2024.

Marra, G., Dumančić, S., Manhaeve, R., and De Raedt, L. From Statistical Relational to Neurosymbolic Artificial Intelligence: A survey. *Artificial Intelligence*, 328:104062, 2024.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Granada, Spain, 2011.

Prud'homme, C. and Fages, J.-G. Choco-solver: A Java Library for Constraint Programming. *Journal of Open Source Software*, 7(78):4708, 2022.

Roychowdhury, S., Diligenti, M., and Gori, M. Regularizing Deep Networks with Prior Knowledge: A Constraint-based Approach. *Knowledge-Based Systems*, 222:106989, 2021.

Smet, L. D., Venturato, G., Raedt, L. D., and Marra, G. Relational Neurosymbolic Markov Models. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, Philadelphia, USA, 2025. To appear.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E. D., Kurakin, A., and Li, C. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems 33*, Virtual Event, 2020.

Sun, R. *Integrating Rules and Connectionism for Robust Commonsense Reasoning*, pp. 273. John Wiley & Sons, Inc., 1994.

Tao, L., Huang, Y.-X., Dai, W.-Z., and Jiang, Y. Deciphering Raw Data in Neuro-Symbolic Learning with Provable Guarantees. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, BC, Canada, 2024.

Towell, G. G. and Shavlik, J. W. Knowledge-based Artificial Neural Networks. *Artificial Intelligence*, 70(1):119–165, 1994.

Valiant, L. G. A Theory of The Learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

van Krieken, E., Acar, E., and van Harmelen, F. Analyzing Differentiable Fuzzy Logic Operators. *Artificial Intelligence*, 302:103602, 2022.

Vapnik, V. N. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

Verreet, V., De Raedt, L., and Bekker, J. Modeling PU Learning Using Probabilistic Logic Programming. *Machine Learning*, 113(3):1351–1372, 2023.

Wang, J., Deng, D., Xie, X., Shu, X., Huang, Y.-X., Cai, L.-W., Zhang, H., Zhang, M.-L., Zhou, Z.-H., and Wu, Y. TacValuer: Knowledge-based Stroke Evaluation in Table Tennis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3688–3696, Virtual Event, Singapore, 2021.

Wang, K., Tsamoura, E., and Roth, D. On Learning Latent Models with Multi-instance Weak Supervision. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA, USA, 2023.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Van den Broeck, G. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5502–5511, Stockholm, Sweden, 2018.

Yang, X.-W., Wei, W.-D., Shao, J.-J., Li, Y.-F., and Zhou, Z.-H. Analysis for abductive learning and neural-symbolic reasoning shortcuts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56524–56541, Vienna, Austria, 2024.

Yang, Z., Ishay, A., and Lee, J. NeurASP: Embracing Neural Networks into Answer Set Programming. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 1755–1762, Yokohama, Japan, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhou, Z.-H. Abductive Learning: Towards Bridging Machine Learning and Logical Reasoning. *Science China Information Sciences*, 62(7):76101, 2019.

Appendix

The appendix is structured as follows.

- Appendix A contains proofs omitted in the main paper, because of the space limit.
- Appendix B contains more details and additional experiments.

A. Proofs

In this section, we give the proofs that are omitted in the main text. For the convenience of the reader, we re-state the assumptions, lemmas, propositions, and theorems in the appendix again.

A.1. Proof of Theorem 3.3

Theorem 3.3. *A minimizer of R_{PNL} or R_{ABL} is also a minimizer of R_{NeSy}. For each surrogate $R_s \in \{R_{PNL}, R_{ABL}\}$, we have:*

$$\arg\min_{f\in\mathcal{F}} R_s(f) \subseteq \arg\min_{f\in\mathcal{F}} R_{NeSy}(f)\,.$$

Proof. First, we recall the risks as follows:

$$R_{\text{PNL}}(f) = -\mathbb{E}_{(\boldsymbol{x},y)} \log \sum_{\boldsymbol{z}} \mathbb{I}(\boldsymbol{z} \wedge \texttt{KB} \models y) \cdot \Pr\left[\boldsymbol{z} \mid \boldsymbol{x}; f, \texttt{KB}\right]\,,$$
$$R_{\text{ABL}}(f) = -\mathbb{E}_{(\boldsymbol{x},y)} \log \left(\Pr\left[y, \bar{\boldsymbol{z}} \mid \boldsymbol{x}; f, \texttt{KB}\right]\right)\,.$$

To unify the proof, we introduce (9) as a surrogate form:

$$R_{\text{A}^3}(f) = -\mathbb{E}_{(\boldsymbol{x},y)} \log \left(\sum_{\bar{\boldsymbol{z}}\in\mathcal{N}(y)} \Pr\left[y, \bar{\boldsymbol{z}} \mid \boldsymbol{x}; f, \texttt{KB}\right] \right)\,,$$

where the set $\mathcal{N}(y)$ denotes the candidate set satisfying:

$$\forall \boldsymbol{z} \in \mathcal{N},\ \boldsymbol{z} \wedge \texttt{KB} \models y\,.$$

In most practical scenarios, due to time or computational constraints, this set may not include all possible candidates. Furthermore, we reformulate the PNL risk using a union-based representation:

$$\begin{aligned} R_{\text{PNL}}(f) &= -\mathbb{E}_{(\boldsymbol{x},y)} \log \sum_{\boldsymbol{z}\in A(y)} \mathbb{I}(\boldsymbol{z} \wedge \texttt{KB} \models y) \cdot \Pr\left[\boldsymbol{z} \mid \boldsymbol{x}; f, \texttt{KB}\right] \\ &= -\mathbb{E}_{(\boldsymbol{x},y)} \log \left(\sum_{\boldsymbol{z}\in A(y)} \Pr\left[y, \boldsymbol{z} \mid \boldsymbol{x}; f, \texttt{KB}\right] \right)\,. \end{aligned}$$

Consequently, R_{A^3} emerges as the most flexible surrogate form. By adjusting the size of the candidate set, we can interpolate between the ABL risk and the PNL risk. Thus, it suffices to prove that for any candidate set $\mathcal{N}(y)$, R_{A^3} achieves the desired objective.

Since the following properties hold:

- For any $f \in \mathcal{F}$, the risk $R_{\text{A}^3}(f) \geq 0$.
- For the labeling function g, the risk $R_{\text{A}^3}(g) = 0$.

Therefore, the minimum achievable value of this risk is strictly 0. For any $f^* \in \arg\min_{f\in\mathcal{F}} R_{\text{A}^3}(f)$, we have $R_{\text{A}^3}(f^*) = 0$, which implies that, for a fixed candidate set $\mathcal{N}(y)$ and any $(\boldsymbol{x}, y)$:

$$\begin{aligned}&\sum_{\bar{\boldsymbol{z}}\in\mathcal{N}(y)} \Pr\left[y, \bar{\boldsymbol{z}} \mid \boldsymbol{x}; f^*, \texttt{KB}\right]\\ =&\sum_{\bar{\boldsymbol{z}}\in\mathcal{N}(y)} \mathbb{I}(\bar{\boldsymbol{z}} \wedge \texttt{KB} \models y) \cdot p_{\theta^*}(\bar{\boldsymbol{z}} \mid \boldsymbol{x}) = 1\,.\end{aligned}$$

Consequently, any $\bar{z}$ predicted by the learning model f^* with a probability greater than zero will satisfy the knowledge base, i.e., $\bar{\boldsymbol{z}} \wedge \texttt{KB} \models y$. This ensures that the NeSy risk $R_{\text{NeSy}}(f) = -\mathbb{E}_{(\boldsymbol{x},y)}\left[\mathbb{I}\left(f^*(\boldsymbol{x}) \wedge \texttt{KB} \not\models y\right)\right]$ is also zero. Since the NeSy risk should also be greater or equal to zero, which means the hypothesis f^* is a minimizer of NeSy risk. Hence, the proof is complete. □

A.2. Proof of Proposition 4.2

Proposition 4.2. *For an ambiguous NeSy task $\mathcal{T}$, if the hypothesis space $\mathcal{F}$ is sufficiently complex (e.g., capable of shattering the task), there exists f^* that minimizes R_{NeSy} but does not minimize the $R_{0/1}$.*

Proof. By the definitions of R_{NeSy} and $R_{0/1}$, we have:

$$R_{\text{NeSy}}(f) = \mathbb{E}_{(\boldsymbol{x},y)}\left[\mathbb{I}\left(f(\boldsymbol{x}) \wedge \texttt{KB} \not\models y\right)\right],$$

and, thus,

$$R_{0/1}(f) = \mathbb{E}_{x,z}\left[\mathbb{I}(f(x) \neq z)\right].$$

Since $\mathcal{T}$ is ambiguous, i.e., there exists a $y \in \mathcal{Y}$ such that $|A(y)| \geq 2$, we assume, without loss of generality, a sample pair $(\boldsymbol{x}_0, y_0) \in (\mathcal{X}^m, \mathcal{Y})$ such that $\{\boldsymbol{z}_1, \boldsymbol{z}_2\} \subseteq A(y_0)$.

Given that the hypothesis space $\mathcal{F}$ is sufficiently complex to shatter the task, we assume the existence of two hypotheses f_1 and f_2 that yield identical correct predictions for all inputs except $\boldsymbol{x}_0$:

$$\begin{cases} f_1(\boldsymbol{x}_0) = \boldsymbol{z}_1, \quad f_2(\boldsymbol{x}_0) = \boldsymbol{z}_2 & \text{if } \boldsymbol{x} = \boldsymbol{x}_0, \\ f_1(\boldsymbol{x}) = f_2(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

By definition, as f_1 and f_2 yield identical predictions except at $\boldsymbol{x}_0$, we have $R_{\text{NeSy}}(f_1) = R_{\text{NeSy}}(f_2)$. However, since $\boldsymbol{z}_1 \neq \boldsymbol{z}_2$, there exists at least one index $k \in [m]$ such that $(\boldsymbol{z}_1)_k \neq (\boldsymbol{z}_2)_k$. Thus, by the definition of $R_{0/1}$, we observe that $R_{0/1}(f_1) \neq R_{0/1}(f_2)$, as at the sample $(\boldsymbol{x}_0)_k$, they produce two distinct recognition results.

In this scenario, even if f_1 represents the underlying ground truth mapping function, it is indistinguishable from f_2, as both achieve zero risk under the optimized objective R_{NeSy}. This concludes that $(R_{\text{NeSy}} \to 0) \not\Rightarrow (R_{0/1} \to 0)$. □

A.3. Proof of Lemma 4.6

Lemma 4.6. *For a NeSy task $\mathcal{T}$, if the DCSP solution disagreement $d = 0$, then the NeSy risk is equivalent to the concept risk. Formally, for any $f \in \mathcal{F}$:*

$$R_{NeSy}(f) \to 0 \iff R_{0/1}(f) \to 0.$$

Proof. We first prove the direction:

$$(R_{\text{NeSy}} \to 0) \Leftarrow (R_{0/1} \to 0)\,.$$

This is evident because, as $R_{0/1}$ approaches zero, f must correctly classify all input-output pairs, which consequently drives R_{NeSy} to zero as well.

Next, we prove the direction: $(R_{\text{NeSy}} \to 0) \Rightarrow (R_{0/1} \to 0)$. Equivalently, we prove the contrapositive:

$$(R_{0/1} \not\to 0) \Rightarrow (R_{\text{NeSy}} \not\to 0)\,.$$

Suppose $R_{0/1} \not\to 0$. Then, there exist integers $i, j \in [L]$ with $i \neq j$ such that f misclassifies elements of the set

$$\langle x \rangle_i = \{x \mid x \in \mathcal{X}, g(x) = i\}$$

as belonging to class j.

Recall that the DCSP of $\mathcal{T}$ has a unique solution, which ensures that the correct labels are unambiguous. As the training set size grows, there must exist a sample $(\boldsymbol{x}, y) \in (\mathcal{X}^m, \mathcal{Y})$ such that set$(\boldsymbol{x}) \cap \langle x \rangle_i \neq \emptyset$ and $f(\boldsymbol{x}) \notin A(y)$. This implies that $R_{\text{NeSy}} \not\to 0$.

Thus, by proving both directions, we complete the proof. □

A.4. Proof of Theorem 4.9

First, we recall that the learnability analysis depends on two assumptions.

Assumption 4.7. (Finite Cardinality) *The set of possible concept sequences, $\mathcal{B} = \bigcup_{y \in \mathcal{Y}} A(y)$, has finite cardinality.*

Assumption 4.8. (Non-vanishing Probability) *The sampling process is controlled by a distribution $P(Z)$, and for any concept sequence $\boldsymbol{z} \in \mathcal{B}$, the probability of being sampled is at least a small positive constant κ.*

Based on the assumptions, we first prove the below lemma, which states the sample complexity under the learnable case, when the hypothesis space is restricted hypotheses space.

Lemma A.1. *Consider a NeSy task $\mathcal{T}$ with above assumptions and $d = 0$. By applying empirical risk minimization, the hypothesis $\hat{f} = \arg\min_{f \in \mathcal{F}^*} \hat{R}_{NeSy}(f)$ ensures that $R_{0/1}(\hat{f}) \leq \epsilon$ for any $\epsilon > 0$, provided that the training size N satisfies the inequality:*

$$N > \frac{1}{\kappa} \cdot \log\left(\frac{|\mathcal{B}|}{\epsilon}\right).$$

Proof. Recall that empirical risk minimization, based on the neuro-symbolic risk $\hat{R}_{\text{NeSy}}$, corresponds to solving a derived constraint satisfaction problem over the restricted hypothesis space $\mathcal{F}^*$. By lemma 4.6, if the training set includes all possible concept sequences, the minimum value of R_{NeSy} becomes zero. This ensures that $R_{0/1}$ also attains a value of zero. Therefore, it is crucial to analyze the sampling process of the training data.

Let Q denote the event that *not all concept sequences are sampled in the training data*. Therefore, we conclude that the true risk is bounded by the probability of event Q:

$$R_{0/1}(\hat{f}) \leq \Pr[Q].$$

To bound $R_{0/1}$, it suffices to bound $\Pr[Q]$. For any individual concept sequence $\boldsymbol{z}_i$, the probability that it is not sampled after N draws is given by:

$$(1 - p_i)^N \leq (1 - \kappa)^N .$$

Applying the union-bound inequality, we derive:

$$\Pr[Q] \leq |\mathcal{B}| (1 - \kappa)^N .$$

Since $(1 - x) \leq \exp(-x)$ holds for $x \geq 0$, we can further bound $R_{0/1}(\hat{f})$ as follows:

$$R_{0/1}(\hat{f}) \leq \Pr[Q] \leq |\mathcal{B}| \exp(-N \cdot \kappa).$$

Given that $R_{0/1}(\hat{f}) \leq \epsilon$, it follows that:

$$N \geq \frac{1}{\kappa} \cdot \log\left(\frac{|\mathcal{B}|}{\epsilon}\right).$$

This completes the proof of the proposition. □

Theorem 4.9. *For a neuro-symbolic task $\mathcal{T}$ with a restricted hypothesis space $\mathcal{F}^*$ exhibiting the clustering property, learnability is determined by the conditions:*

- *If the derived constraint satisfaction problem has a unique solution, the task is* learnable. *Specifically, the concept error is bounded by ϵ, given that the sample size N satisfies the following condition:*

$$N > \frac{1}{\kappa} \cdot \log\left(|\mathcal{B}|/\epsilon\right) .$$

- *Otherwise, the task is* unlearnable.

Proof. The proof is divided into two parts:

1. If the disagreement d equals zero, the task is *learnable*, and the sample complexity is $\mathcal{O}(\frac{1}{\kappa} \cdot \log(|\mathcal{B}|/\epsilon))$.
2. If the disagreement d is greater than zero, the DCSP solution space contains at least two distinct solutions, making the task *unlearnable*.

The first part follows directly from lemma A.1. Hence, we focus on proving the second part by contradiction.

If the DCSP has multiple solutions, there exists $(\boldsymbol{x}, y) \in (\mathcal{X}^m, \mathcal{Y})$ such that two distinct concept sequences $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are valid, i.e., $\boldsymbol{z}_1 \wedge \texttt{KB} \models y$ and $\boldsymbol{z}_2 \wedge \texttt{KB} \models y$.

Since both $\hat{f}_1(\boldsymbol{x}) = \boldsymbol{z}_1$ and $\hat{f}_2(\boldsymbol{x}) = \boldsymbol{z}_2$ are valid solutions for $(\boldsymbol{x}, y)$, and $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are distinct, it follows that their true risks cannot be simultaneously zero. Thus, at least one of them must have a $R_{0/1}$ greater than zero. Without loss of generality, assume that $R_{0/1}(\hat{f}_1) = \epsilon_0 > 0$.

Since both $\hat{f}_1$ and $\hat{f}_2$ achieve the minimal NeSy risk (which is zero), it is impossible to distinguish between them using learning techniques or by adding more data. Consequently, there is no integer N_ϵ such that for any $0 < \epsilon < \epsilon_0$, $R_{0/1}(\hat{f}) < \epsilon$ holds when $N \geq N_\epsilon$. This implies that the task is unlearnable.

Combining both parts completes the proof. □

A.5. Proof of Theorem 4.10

Theorem 4.10. *The average error $\mathcal{E}^*$ is bounded by:*

$$\mathcal{E}^* \leq \frac{d}{L} .$$

Proof. Recall that the DCSP solution disagreement d is given by $d = L - \text{Union}(\mathcal{S})$, where $\text{Union}(\mathcal{S})$ represents the common assignments among the solutions. Since the restricted hypothesis space ensures that instances with the same assigned label correspond to the same concept and vice versa. In the worst case, errors occur in at d classes, so the maximum true risk is $\max_{f \in \mathcal{F}^*_{\text{ERM}}} R_{0/1}(f) = d/L$. Thus, the average error is bounded by: $\mathcal{E}^* \leq \max_{f \in \mathcal{F}^*_{\text{ERM}}} R_{0/1}(f) = d/L$. □

B. Experiments

We first introduce the experimental details, including data preparation, model setup, optimizer configurations, hyperparameters, and implementation details. After that, we present experiments omitted from the main context due to space constraints.

B.1. Experiment Details

The raw datasets are based on MNIST (Deng, 2012), KMNIST (Clanuwat et al., 2018), CIFAR-10 (Krizhevsky, 2009), and SVHN (Netzer et al., 2011). For MNIST-style datasets, the learning model is based on LeNet (LeCun & Bengio, 1998); other datasets use ResNet (He et al., 2016). The results were obtained using an Intel Xeon Platinum 8538 CPU and an NVIDIA A100-PCIE-40GB GPU on an Ubuntu 20.04 platform.

B.1.1. Preparing data and model

The construction of datasets is heavily based on algorithmic operations; thus, we rely on digit indices mapping from class indices to digit indices. After that, different knowledge bases require different rules. Here, we base our approach on ABLKit (Huang et al., 2024)[1] and the code of He et al. (2024b)[2]. During the dataset construction process, we control the sample size by re-sampling data until the sequence size exceeds a threshold, denoted as `sample_size`. For figure 3, the `sample_size` is set to $30,000$, while for ensemble experiments it is set to $120,000$; other values are specified in the respective plots. The reasoning model employs abductive reasoning, implemented using a cache-based search program (Huang et al., 2024).

For an example of addition, the knowledge base is programmed as follows:

```
class add_KB(KBBase):
  ...
  def logic_forward(self, nums):
    nums1, nums2 = split_list(nums)
    return digits_to_number(nums1) + digits_to_number(nums2)
```

Figure 5: Example of addition knowledge base with Python program form.

For the modular addition task, the knowledge base is more complex:

```
class Mod_KB(KBBase):
  ...
  def logic_forward(self, lsts):
      nums1, nums2, mod = parse_nums_and_mods(lsts)
      nums1, nums2 = digits_to_number(nums1), digits_to_number(nums2)
      return (nums1 + nums2) % mod
```

Figure 6: Example of modular addition knowledge base with Python program form.

B.1.2. Implementation details

For all experiments, the random seeds are set to $\{2023, 2024, 2025, 2026, 2027\}$ for repeating five times. To ensure the clustering property depicted in the definition of the restricted hypothesis space, the learning models are pre-trained. For LeNet5, we use self-supervised methods, with the weights available in the supplementary materials. For ResNet50, we load the pre-trained weights from the official PyTorch library, named `ResNet50_Weights.IMAGENET1K_V2`, and replace the last linear layer with `Linear(2048, 10)`.

Optimization configurations All experiments use AdamW, a weight-decay variant of Adam (Kingma & Ba, 2015), as the optimizer, with a learning rate of 0.0015 and betas set to $(0.9, 0.99)$. The batch size is set to 256, and unless otherwise noted, the number of epochs is set to 10. The loss function used for optimization is cross-entropy, with further details

[1] https://github.com/AbductiveLearning/ABLkit
[2] https://github.com/Hao-Yuan-He/A3BL

Algorithm 1 DCSP Solution

Require: NeSy task $\mathcal{T}$ and training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
1: $\mathsf{V}, \mathsf{D}, \mathsf{C} \leftarrow \{V_i\}_{i=1}^{L}, \{D_i = \mathcal{Z}\}_{i=1}^{L}, \{\}$ ▷ Initialize the CSP triple
2: **for** $i = 1 \ldots N$ **do**
3: $\quad \mathsf{C} \leftarrow \mathsf{C} \cup \{\mathsf{V}(\boldsymbol{x}_i) \wedge \mathtt{KB} \models y_i\}$ ▷ Initialize the constraints
4: **end for**
5: $\mathcal{S} \leftarrow \textsc{SolveCSP}(\mathsf{V}, \mathsf{D}, \mathsf{C})$ ▷ Call the CSP solver
6: $d \leftarrow L - \text{Union}(\mathcal{S})$
7: **return** $\mathcal{S}, d$

available in the ABLKit code.

B.2. Additional Experiments

The additional experiments include setups using raw datasets not covered in the main text, specifically the CIFAR-10 and SVHN cases. Additionally, beyond the empirical analysis on the surrogate (9), here we refer to this as A^{3}BL (He et al., 2024b), we also include an analysis of ABL and PNL to ensure a comprehensive evaluation.

B.2.1. Observations of the structure of DCSP solution space

The configurations of the modular addition task and their ensembles are illustrated in figure 7. These configurations were computed using algorithm 1 with the open-source library Choco (Prud'homme & Fages, 2022). Through the investigation of the modular addition task, we observe that: *the number of DCSP solutions is highly related to DCSP solution disagreement; however, this relationship is not monotonic.* Specifically, even a small number of DCSP solutions can result in high disagreement, as observed in the modular addition task with base $k = 10$.

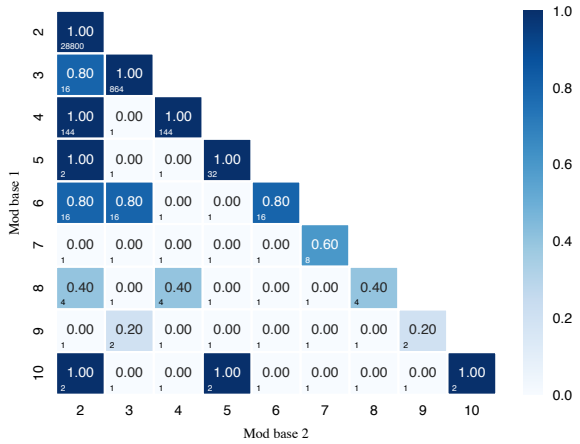
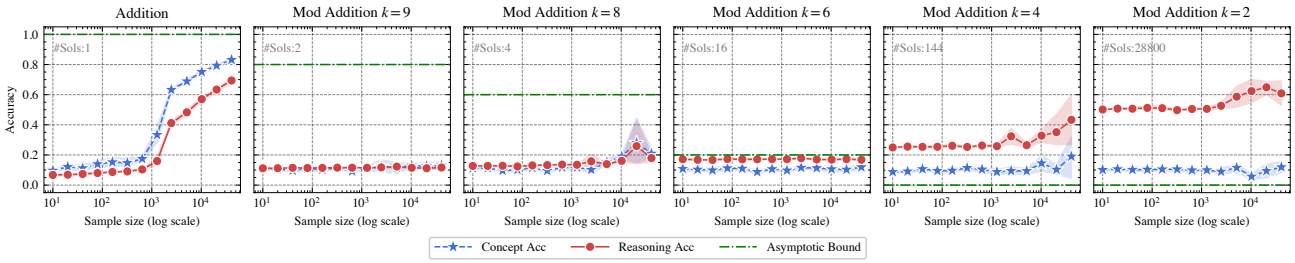


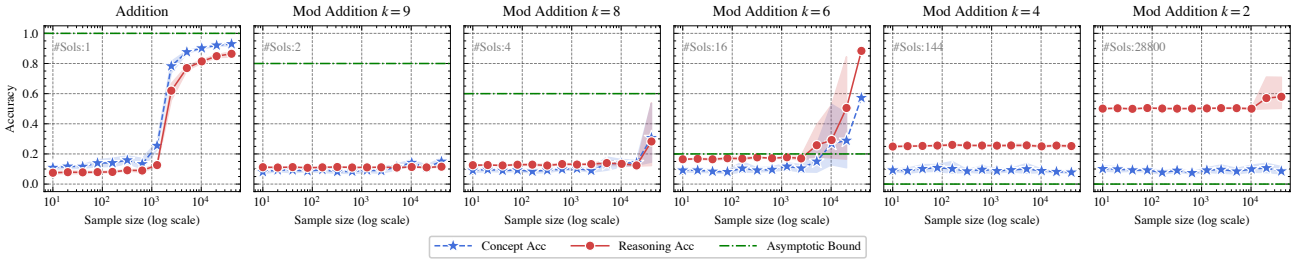
Figure 7: Configurations of modular additions and their ensembles. The center value represents the ratio of disagreement d to concept size L, while the number of DCSP solutions is shown at the bottom-left.

B.2.2. Impact of DCSP solution disagreement

In figure 8, we present results under the same settings but using different raw datasets, specifically CIFAR-10 and SVHN, as shown in figure 8. As illustrated in the figure, the learnable case follows a trend similar to that in figure 2. However, in the unlearnable case, optimization becomes significantly more challenging due to high conflicts among valid DCSP solutions. While one might suspect that this issue stems from the specific surrogate used, applying the same settings to ABL and PNL produces similar results (cf. figure 9 and figure 10 respectively), confirming the generality of this observation.

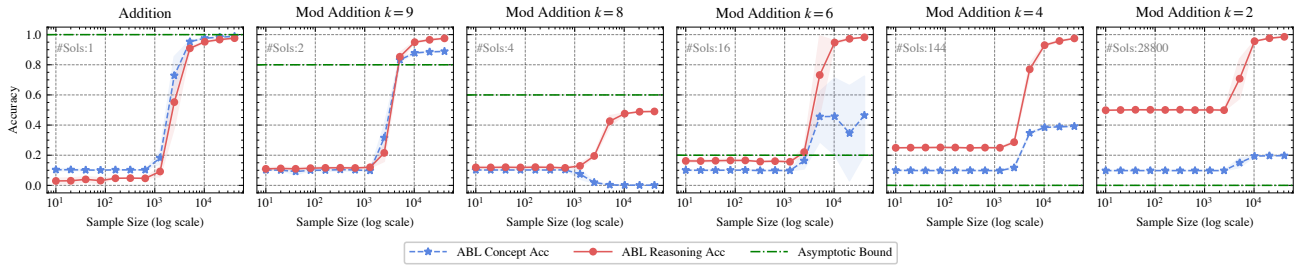


(a) CIFAR-10

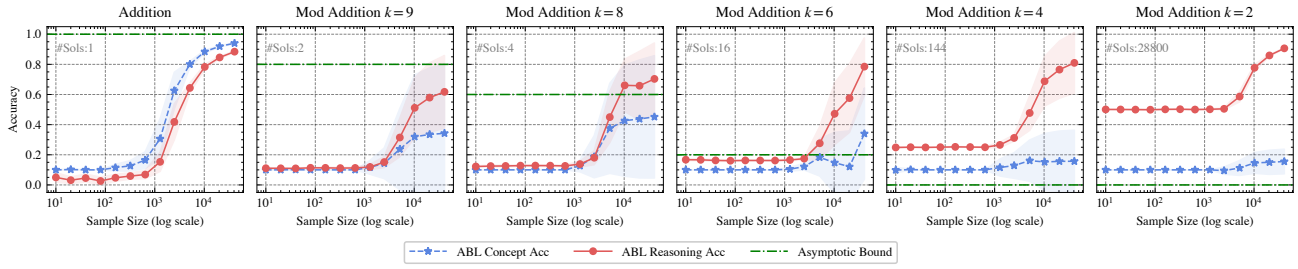


(b) SVHN

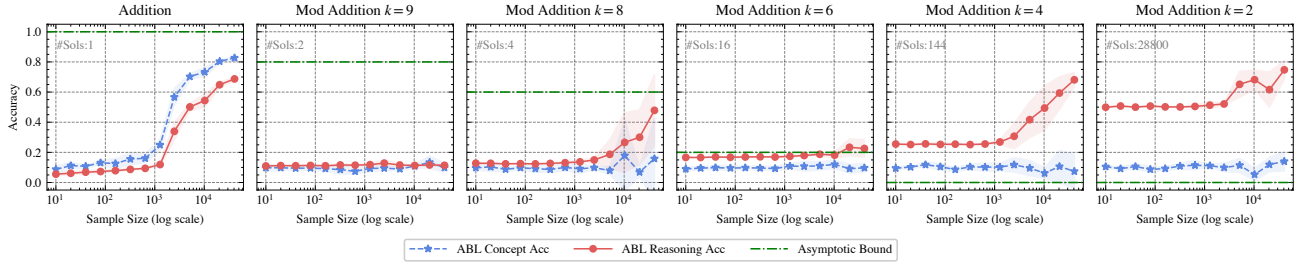
Figure 8: *Accuracies* versus *sample size* for different NeSy tasks of A^{3}BL. The shadowed area denotes the standard error. The number of the DCSP solutions (#Sols) is shown at the top left of each plot. The asymptotic bound (green line) from theorem 4.10 indicates that concept accuracy should exceed this bound as the sample size grows.



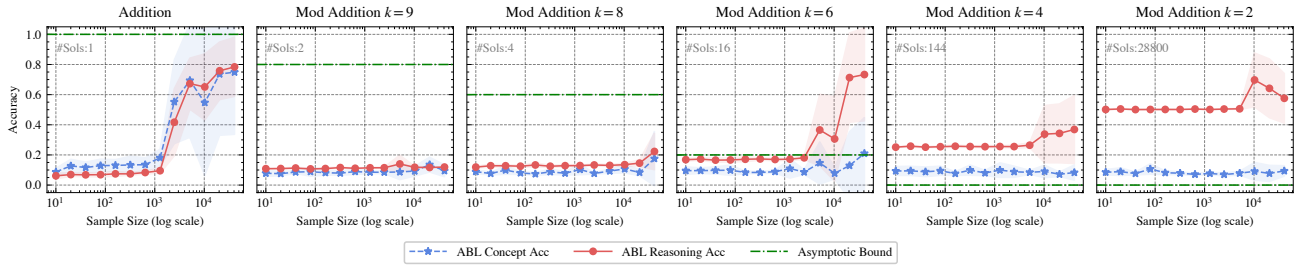
(a) MNIST



(b) KMNIST

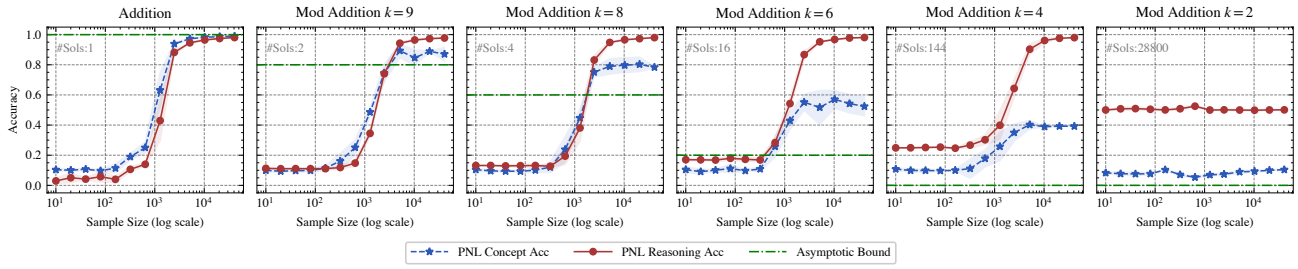


(c) CIFAR-10

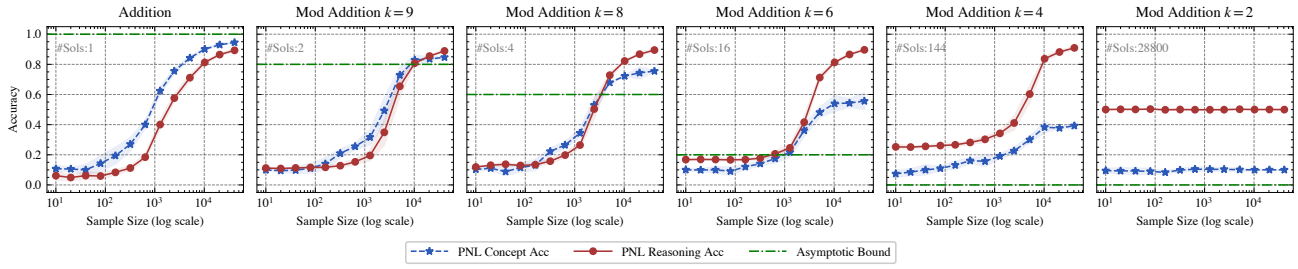


(d) SVHN

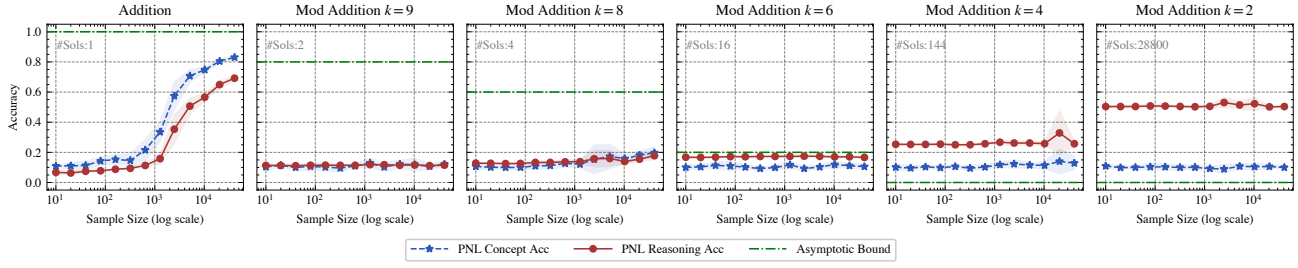
Figure 9: *Accuracies* versus *sample size* for different NeSy tasks of ABL. The shadowed area denotes the standard error. The number of the DCSP solutions (#Sols) is shown at the top left of each plot. The asymptotic bound (green line) from theorem 4.10 indicates that concept accuracy should exceed this bound as the sample size grows.



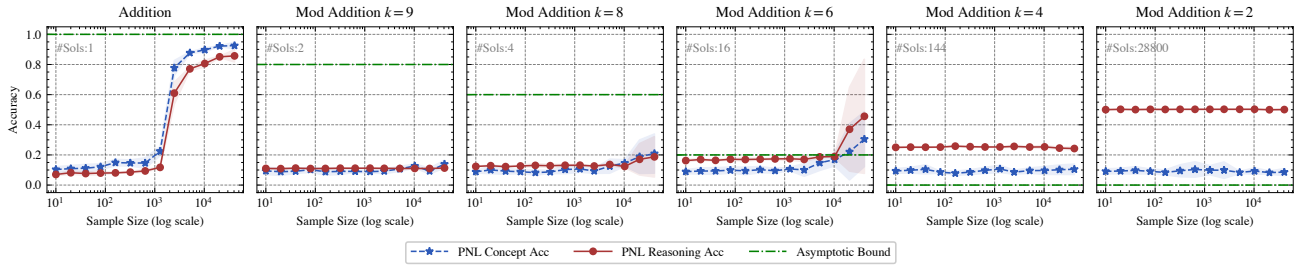
(a) MNIST



(b) KMNIST



(c) CIFAR-10



(d) SVHN

Figure 10: *Accuracies* versus *sample size* for different NeSy tasks of PNL. The shadowed area denotes the standard error. The number of the DCSP solutions (#Sols) is shown at the top left of each plot. The asymptotic bound (green line) from theorem 4.10 indicates that concept accuracy should exceed this bound as the sample size grows.

B.2.3. Ensemble of unlearnable NeSy tasks

Here we present additional combinations of unlearnable NeSy tasks. In figure 11, we provide an overview of all ensemble combinations using a heatmap. After that, we present a more fine-grained analysis. In figure 12 and figure 13, we show cases where the ensemble approach fails and succeeds respectively.

Figure 11: *Heatmaps* of ensemble mod addition tasks. Left: reasoning accuracy for different ensembles of mod bases; Right: concept accuracy for different ensembles of mod bases. The bottom-left corner of each cell shows the number of DCSP solutions.

By observing the experiments, we find that adopting the ensemble perspective can enrich the benchmark diversity in the NeSy field (e.g., `rsbench`, Bortolotti et al. 2024), providing a clear and controllable methodology to achieve this.

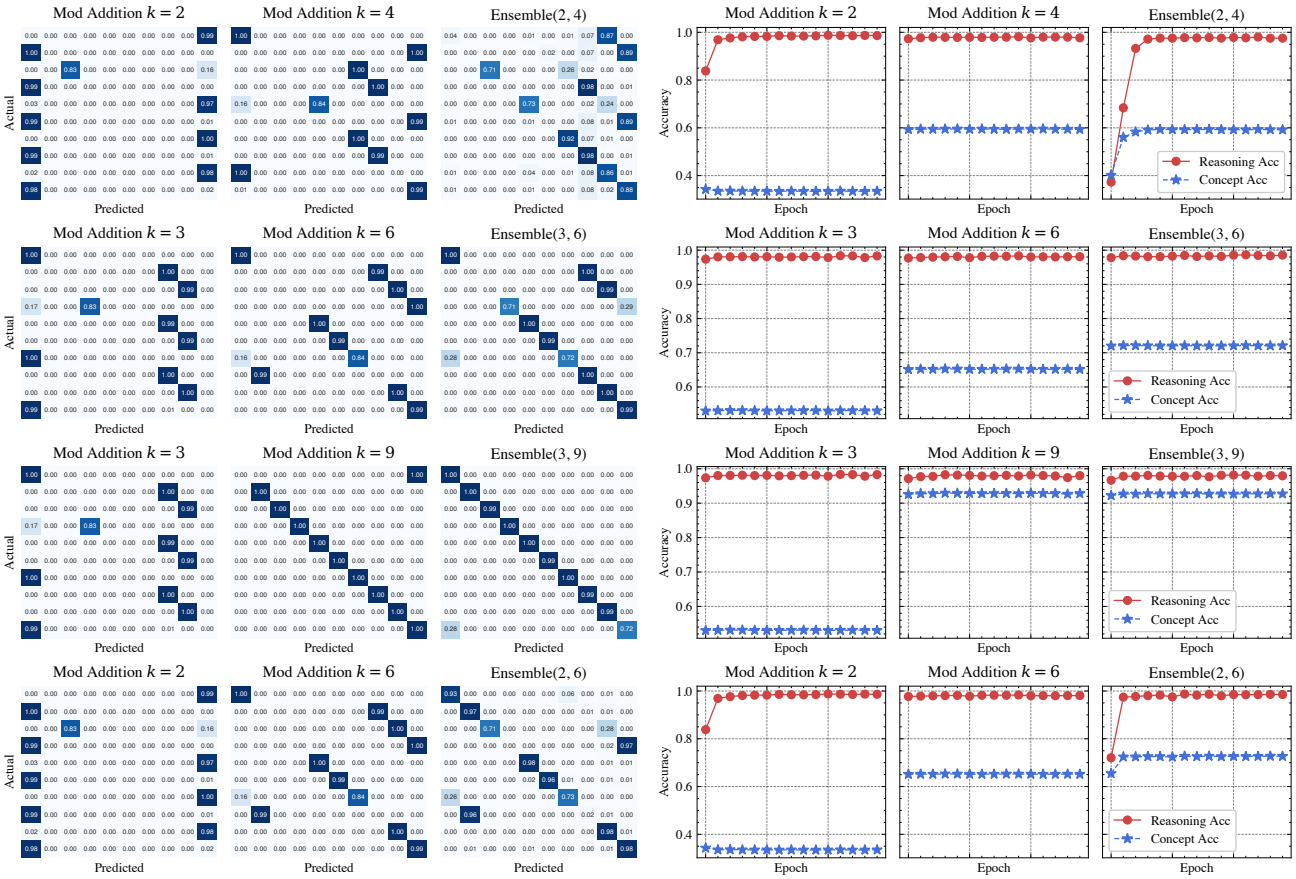


Figure 12: *Ensemble* of unlearnable NeSy tasks, *failed* case. The left shows confusion matrices, and the right displays accuracy curves. After the ensemble, the tasks are still unlearnable.

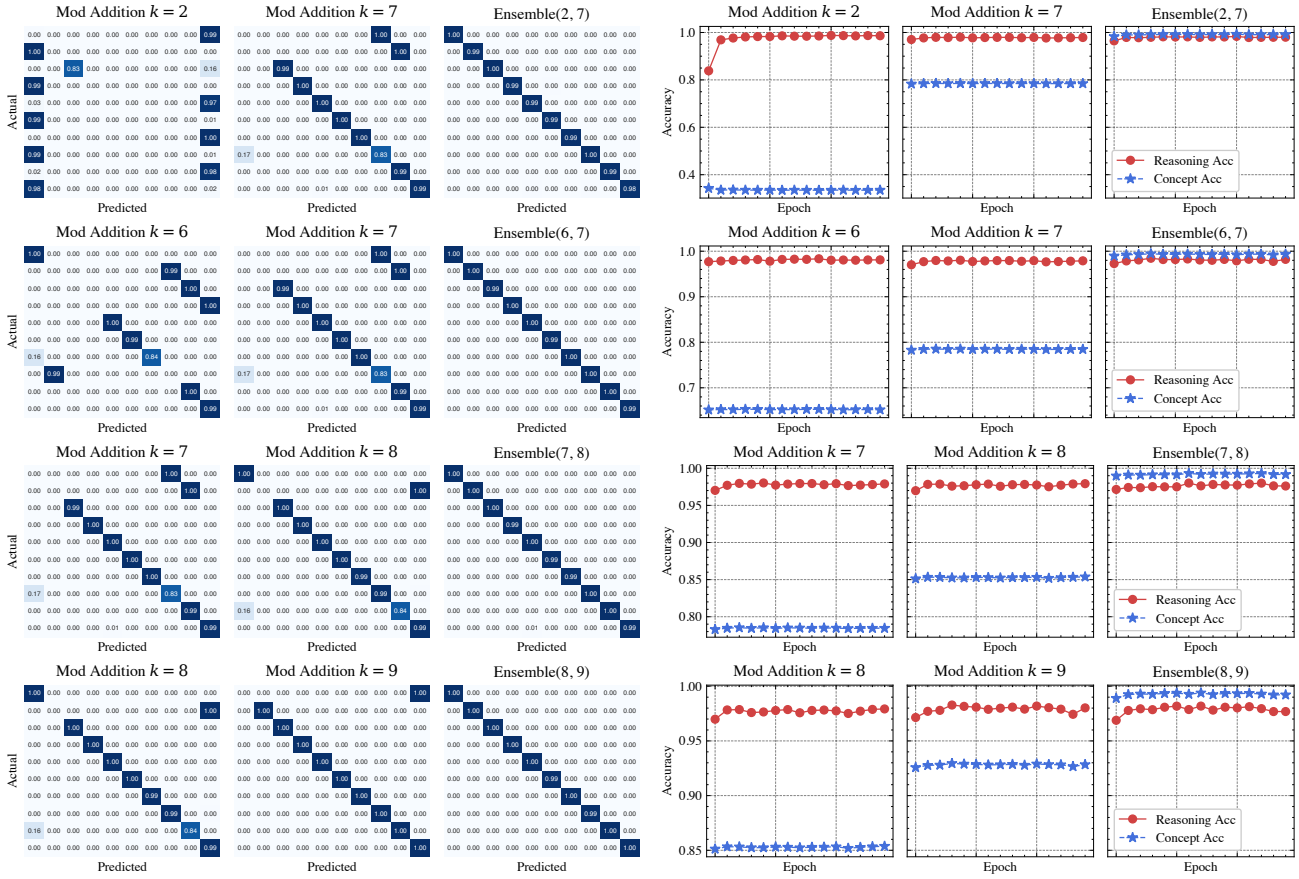


Figure 13: *Ensemble* of unlearnable NeSy tasks, *successed* case. The left shows confusion matrices, and the right displays accuracy curves. After the ensemble, the tasks become learnable.