

Does Reasoning Mitigate Backdoor Attacks? A Neuro-Symbolic Perspective

Marco Antonio Corallo[✉]
Örebro University
Örebro, Sweden
marco-antonio.corallo@oru.se

Mauro Conti
University of Padua
Padua, Italy
Örebro University
Örebro, Sweden
mauro.conti@unipd.it

Andrea Agiollo
Delft University of Technology
Delft, The Netherlands
a.agiollo-1@tudelft.nl

Alberto Giarretta
Örebro University
Örebro, Sweden
Alberto.Giarretta@oru.se

Abstract

Neuro-Symbolic (NeSy) AI has recently emerged as a novel paradigm to enable trustworthy AI, aiming at integrating sub-symbolic neural perception with grounded symbolic reasoning. The neuro-symbolic integration process that characterizes these models has been proven beneficial to achieve more transparent, explainable and efficient AI systems. Meanwhile, their properties under adversarial settings have been overlooked being frequently deemed robust-by-design. However, the neural-symbolic integration process they leverage constitutes an additional layer of complexity that may provide an attack entry-point. Therefore, in this paper, we claim that an in-depth investigation of the adversarial robustness of NeSy models is necessary and provide the first systematic evaluation of backdoor attacks against NeSy. To this end, we compare the most popular NeSy framework, namely DeepProbLog, against baseline neural networks across a total of eight backdoor settings and four reasoning tasks. Our experimental results show that while NeSy models are indeed more robust than their neural counterpart on average, their robustness vastly depend on the strictness of the reasoning process being enforced and its compatibility with the chosen adversarial target. The source code to reproduce our experiments is made available at <https://github.com/marcoantoniorcorallo/NeSy-Backdoor>.

Keywords

Machine Learning Security, Neuro-Symbolic AI, Trustworthy ML

1 Introduction

Neural networks (NNs) have achieved super-human level of performance across a variety of tasks [16, 23]. Nevertheless, their increased deployment has exposed several of their limitations, such as data-hungerness, limited transparency and susceptibility to adversarial manipulations [47, 48]. These characteristics hinder their application in safety-critical domains, where full control over the decision process is a must. To address these limitations, Neuro-Symbolic (NeSy) AI has emerged as a novel paradigm aiming to combine the representation-learning and perceptual capabilities of neural networks with the structured knowledge and reasoning mechanisms traditionally associated with symbolic AI [10, 30].

Recently, several NeSy architectures have been proposed, each characterized by a different way of incorporating the desired symbolic knowledge, either through logical constraints, differentiable reasoning operators, semantic regularization objectives, or others [26, 41, 43]. These architectures have been commonly proven to be associated with several desirable properties, including improved interpretability, data efficiency and consistency with domain knowledge [2, 3, 12]. An underlying root cause for such properties is to be found in the symbolic constraints restricting the space of admissible predictions and penalizing violations of known relations. This property has also motivated the view that NeSy models may be more robust and trustworthy than their purely neural counterparts, with existing studies reporting benefits under noisy observations and distributional shifts [25, 33]. As such, one of the principal motivations for combining learning and reasoning in NeSy AI is now frequently pointed as their increased robustness.

Robustness under naturally occurring uncertainty, however, does not imply robustness against an adversary. While, random noise and distribution shifts do not deliberately exploit the internal structure of a model, an adversary can manipulate inputs and training data with the explicit objective of inducing a desired failure. This distinction is particularly relevant for NeSy systems, where the symbolic knowledge may penalize inconsistent outputs, but the neural-symbolic interface introduces an additional attacker target. Indeed, a NeSy model may satisfy the encoded knowledge while relying on incorrect concepts, a phenomenon known as reasoning shortcuts [28]. Therefore, an attacker could theoretically exploit such shortcuts to preserve apparent symbolic consistency while compromising the model reasoning.

These concerns raise a fundamental question: *does the integration of symbolic knowledge make NeSy models inherently more resistant to adversarial attacks, or does it merely alter the way in which attacks affect the model?* Despite the increasing emphasis placed on NeSy AI, their adversarial robustness remains comparatively underexplored. In particular, the robustness benefits attributed to symbolic integration are often inferred from the ability of symbolic knowledge to regularize neural learning [4, 13]. However, such evidence is insufficient for determining whether symbolic components remain effective when the training process is deliberately manipulated as hinted by Agiollo and Omicini [1].

In this work, we investigate this question in the context of backdoor attacks. A backdoor attack aims to embed a hidden, attacker-controlled behavior into a model, commonly by poisoning a small fraction of its training data. The compromised model behaves normally on benign inputs but produces an attacker-selected prediction when a predefined trigger is present [6, 15]. Backdoor attacks provide a particularly informative setting for evaluating NeSy robustness as they allow to investigate if symbolic knowledge can operate as an effective structural safeguard preventing trigger learning. Therefore, we present a systematic evaluation of NeSy models against backdoor attacks and compare their behavior with that of matched neural counterparts. Our experimental design controls for the underlying neural architecture, training data, and optimization settings, thereby isolating the effect of symbolic knowledge integration on backdoor robustness. We consider the most popular NeSy framework, namely DeepProbLog [26], against baseline NNs across a total of eight backdoor settings and four reasoning tasks.

Our experimental evaluation shows that the robustness of NeSy models against backdoor attacks strongly depends on the interaction between the learned trigger and the symbolic reasoning process. The reasoning layer can substantially improve robustness in targeted scenarios by preventing adversarial predictions that violate the encoded logical constraints, while it provides only limited protection against untargeted attacks that aim only at disrupting the model prediction. To analyze more in detail these phenomena, we introduce the notion of *reachability* of a backdoor target and show that NeSy models robustness against backdoors is entirely determined by the compatibility between the attacker-selected target and the symbolic knowledge enforced in the NeSy model. Overall, these findings demonstrate that symbolic reasoning alone cannot prevent poisoned perceptual representations from being learned, but can effectively mitigate their impact whenever the adversarial objective is logically inconsistent with the underlying reasoning process. Therefore, this paper motivates broader investigation into the adversarial robustness of NeSy models across architectures, reasoning mechanisms, and threat models.

Contributions. The main contributions of this work are: (i) we provide the first systematic analysis of the robustness of NeSy against adversarial backdoor attacks; (ii) we experimentally evaluate the robustness of NeSy against its neural counterpart across eight backdoor settings and four reasoning tasks; and (iii) we identify the conditions under which symbolic knowledge improves resistance to backdoor attack and characterize the relative failure modes.

Organization: The remainder of this paper is organized as follows. In Section 2, we provide the necessary background on NeSy models and backdoor attacks, while Section 3 presents the related work. Subsequently, we define the threat model and methodology we use to analyze backdoors over NeSy in Sections 4 and 5, respectively. Section 6 provides the obtained experimental results, while Section 7 analyzes when backdoors are logically effective. Lastly, Sections 8 and 9 present the limitations, future directions and conclusions.

2 Background

2.1 Neuro-Symbolic AI

Neuro-symbolic artificial intelligence aims to combine the complementary strengths of deep neural networks and symbolic reasoning systems. Neural networks excel at learning perceptual representations from high-dimensional data such as images and text, whereas symbolic methods provide interpretable reasoning, explicit knowledge representation, and logical inference. Recently, several NeSy frameworks emerged, which differ depending on the way they represent explicit knowledge, combine learning and reasoning, or define their semantics [30]. As such, NeSy encompasses a broad range of models rather than a single architecture. An intuitive categorization of such integration models presented in [10] distinguishes them according to how symbolic knowledge is integrated into their computation. For example, symbolic knowledge may be translated into differentiable constraints that guide the training of a neural model [29, 43], or encoded within a differentiable architecture whose components implement logical structures or operations [34]. Alternatively, neural components may be integrated into an explicit symbolic reasoning system, which may differ depending on the symbolic formalism and inference semantics [26, 41, 45].

For the context of this paper, we focus on this last family of models which are also referred to as concept-based NeSy when their neural components extract task-relevant intermediate concepts from raw inputs [8]. The symbolic component then combines these concepts with domain knowledge to solve the task at hand. For example, a neural network may recognize the digits contained in two images, while symbolic arithmetic rules determine their sum. We focus on concept-based NeSy models because their explicit concept layer defines an interface between neural perception and symbolic reasoning which is commonly assumed to introduce a robustness property against manipulations.

2.1.1 DeepProbLog. We instantiate our NeSy architecture using DeepProbLog [26], which extends the probabilistic logic programming language ProbLog [11] with neural predicates. ProbLog represents uncertainty through probabilistic facts and annotated disjunctions and computes query probabilities over the possible worlds induced by these choices. DeepProbLog preserves these semantics while allowing neural networks to parameterize selected predicates. Given an input x , a neural predicate N_θ produces a distribution over a finite set of symbolic outputs $z_1, \dots, z_K$,

$$N_\theta(x) = [p_\theta(z_1 | x), \dots, p_\theta(z_K | x)], \text{ s.t., } \sum_{k=1}^K p_\theta(z_k | x) = 1, \quad (1)$$

which instantiates an annotated disjunction. In our concept-based setting, these outputs represent intermediate concepts that participate in logical inference together with the program’s facts and rules. To answer a query, DPL grounds the relevant program, evaluates the required neural predicates, and compiles the resulting formula into a tractable arithmetic circuit [24]. During training, query-level gradients are propagated through probabilistic inference (enabling differentiation) to the neural networks, enabling end-to-end learning without direct supervision of the intermediate concepts $(z_1, \dots, z_K)$.

2.2 Backdoor Attacks

Backdoor attacks implant a hidden, attacker-defined behavior into a machine learning model while preserving its performance on benign inputs. A backdoored model behaves normally when the trigger is absent but produces a maliciously selected output when the trigger is present. Backdoors are commonly introduced by poisoning a subset of the training data [14, 21], although more complex threat models may allow the adversary to control the training procedure or directly modify a pretrained model [19].

Formally, let f_θ be a model trained on a clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, let $\tau : \mathcal{X} \rightarrow \mathcal{X}$ be a trigger-insertion function, and let y_t denote an attacker-selected target label. The adversary aims to learn parameters θ^{bd} such that

$$f_{\theta^{\text{bd}}}(x) = y \quad \text{and} \quad f_{\theta^{\text{bd}}}(\tau(x)) = y_t, \quad (2)$$

for benign samples (x, y) and eligible triggered inputs, respectively. Thus, a successful attack simultaneously maximizes the attack success rate on triggered inputs and preserves the model’s predictive performance on clean data.

In data-poisoning attacks, the adversary constructs a poisoned training set by replacing or augmenting a subset $\mathcal{D}_p \subseteq \mathcal{D}$:

$$\mathcal{D}^{\text{bd}} = (\mathcal{D} \setminus \mathcal{D}_p) \cup \{(\tau(x), \tilde{y}) : (x, y) \in \mathcal{D}_p\}. \quad (3)$$

In *dirty-label* attacks, the attacker changes the labels of poisoned samples, typically by setting $\tilde{y} = y_t$ [9, 14, 21]. In *clean-label* attacks, the original labels are retained, $\tilde{y} = y$, requiring the backdoor to be learned without introducing an explicit inconsistency between the poisoned inputs and their labels [6, 38].

Backdoor attacks also differ in their trigger design. Early methods commonly used fixed, visible patches [14], while subsequent attacks introduced blended patterns [9], hidden or semantic triggers [35], frequency-domain perturbations [40], and input-dependent triggers [36]. These designs pursue different trade-offs among attack effectiveness, trigger stealthiness, and robustness to defenses, but share the objective of associating a trigger condition with attacker-selected model behavior while maintaining performance on benign inputs. However, a comprehensive overview of available backdoor attacks taxonomy is out of the scope of this work and we refer interested readers to [5, 19, 46].

3 Related Work

Neuro-symbolic models are often motivated as more trustworthy than unconstrained neural predictors because they expose semantically meaningful intermediate concepts and use explicit knowledge to constrain their outputs. These benefits, however, depend on whether the learned concepts acquire the semantics assigned to them by the symbolic theory. Marconato et al. [27] first highlighted this limitation in continual learning, showing that a model can satisfy the available knowledge and achieve high predictive accuracy while learning concepts with unintended meanings. This phenomenon was later formalized as a *reasoning shortcut* [28] with subsequent work proposing benchmarks for detecting such shortcuts [7] and analyzing how their occurrence depends on the knowledge base and hypothesis space [44]. Unlike conventional input-level shortcuts, reasoning shortcuts arise at the interface between subsymbolic inputs and symbolic concepts: the concept grounding is incorrect, although the subsequent reasoning remains logically consistent.

While not adversarial, reasoning shortcuts demonstrate that logical consistency alone does not guarantee trustworthy predictions.

Extending the results of Marconato et al. [27], related work has examined NeSy robustness under non-adversarial failures. For example, Rafanelli et al. [33] study naturally occurring distribution shifts and report benefits under data and label noise, while Liang et al. [20] use neuro-symbolic generative models to capture out-of-distribution samples. Complementarily, Manginas et al. [25] develop a formal method for verifying probabilistic NeSy models against bounded perturbations. These works address naturally occurring shifts and concept ambiguity. In contrast, we consider an adaptive adversary that deliberately modifies training to implant trigger-dependent behavior.

At a broader level, Agiollo and Omicini [1] argue that NeSy trustworthiness should be evaluated with respect to the coupling between neural and symbolic components. Similarly, Parmar [32] argue that NeSy systems are not inherently safe and model them as attack surfaces spanning neural perception, symbolic knowledge, reasoning, orchestration, and data layers. These works provide valuable evaluation principles and threat models, but do not experimentally determine how established NeSy frameworks behave under concrete adversarial training-time attacks. Our work complements them with an empirical analysis of how backdoors enter through neural concept learning and propagate through an otherwise unchanged symbolic program.

Finally, Sarvestani et al. [37] and Vilamala et al. [39] design NeSy systems that explicitly improve robustness to adversarial perturbations in traffic-sign recognition and data-stream event processing, respectively. Their results demonstrate how task-specific symbolic knowledge can support robustness, but do not establish whether existing NeSy architectures are robust by design. In particular, their logic and training procedures are constructed as defenses for specific tasks, whereas we evaluate standard concept-based NeSy models without adding robustness mechanisms. Moreover, they consider inference-time perturbations, while we study backdoors implanted through partial control of the training process and activated only by triggered inputs.

4 Threat Model

We consider a *dirty-label training-time data-poisoning* adversary, which represents the most widely adopted model for evaluating backdoor attacks against NN models [15]. We assume the attacker acts as a malicious data provider who can modify a fraction of the training samples before they are supplied to the victim. However, the attacker cannot control the subsequent training procedure and has no knowledge of the victim model architecture or parameters, similarly to [17, 40]. This setting captures realistic scenarios in which data collection, annotation, or dataset curation is outsourced to a third party.

We consider concept-based NeSy tasks in which a neural component extracts intermediate concepts from a raw input and a fixed symbolic program uses these concepts, possibly together with contextual information, to produce the task output. Let $x \in \mathcal{X}$ denote an input (i.e., an image in this paper), $c \in \mathcal{C}$ an optional clean context, $z \in \mathcal{Z}$ the concepts extracted from x , and $y \in \mathcal{Y}$ the task output. A neuro-symbolic predictor can be represented abstractly

as

$$x \xrightarrow{g_\theta} p_\theta(z | x) \xrightarrow{\mathcal{K}(\cdot, c)} y, \quad (4)$$

where g_θ is the neural concept extractor and $\mathcal{K}$ is the symbolic program. The attacker can poison the image and its task-level label, but cannot modify the clean context c , the symbolic program $\mathcal{K}$, or the learning algorithm. Practically speaking, the attack modifies only the input image, leaving the context and symbolic theory unchanged:

$$(x, c, \mathcal{K}) \mapsto (\tau_\phi(x), c, \mathcal{K}). \quad (5)$$

Consequently, the attack can influence symbolic reasoning only indirectly, by changing the concepts predicted from a triggered image. We rely on these assumptions since (i) the attacker is assumed to be capable only of modifying a subset of the training data and in practical applications the input context c may be given a-priori, and (ii) modifying the concept c to c' together with the input x to activate the backdoor would collapse the task to a trivial backdoor attack over the neural component of the DPL model.

Let the clean training dataset be $\mathcal{D} = \{(x_i, c_i, y_i)\}_{i=1}^N$, where c_i is omitted for tasks without an independent context. The adversary selects a poisoning ratio $\rho \in (0, 1)$ and a subset $\mathcal{D}_\rho \subseteq \mathcal{D}$, with $|\mathcal{D}_\rho| = \rho N$. Each selected example is replaced by

$$(x, c, y) \mapsto (\tau_\phi(x), c, \eta(y)), \quad (6)$$

where $\tau_\phi : \mathcal{X} \rightarrow \mathcal{X}$ is an attack-specific trigger transformation and $\eta : \mathcal{Y} \rightarrow \mathcal{Y}$ is the malicious label mapping. The context remains unchanged and the resulting poisoned dataset is

$$\mathcal{D}^{\text{bd}} = (\mathcal{D} \setminus \mathcal{D}_\rho) \cup \{(\tau_\phi(x), c, \eta(y)) \mid (x, c, y) \in \mathcal{D}_\rho\}. \quad (7)$$

In the targeted (i.e., all-to-one) setting, all poisoned examples are assigned an attacker-selected target y_t : $\eta(y) = y_t$. In the untargeted (i.e., all-to-all) setting, η maps each output to a different valid output according to a predetermined mapping. This general definition accommodates both single-label tasks and structured outputs, for which the standard cyclic class mapping may not be applicable.

The victim trains the model normally on $\mathcal{D}^{\text{bd}}$, obtaining parameters θ^{bd} , and the attack has two objectives. First, the backdoored model should preserve its predictive performance on benign inputs: $f_{\theta^{\text{bd}}, \mathcal{K}}(x, c) = y$. Second, applying the trigger at inference time should induce the malicious output: $f_{\theta^{\text{bd}}, \mathcal{K}}(\tau_\phi(x), c) = \eta(y)$. For the neuro-symbolic model, the attacker must achieve this behavior through the compromised neural component while the symbolic reasoning process remains unchanged. In particular, the trigger must alter $p_{\theta^{\text{bd}}}(z \mid \tau_\phi(x))$ so that the predicted concepts, when combined with the clean context by $\mathcal{K}$, produce the desired malicious output. Whether the symbolic program permits this behavior depends on the task and target. We quantify this relationship through the reachability analysis introduced in Section 5.4.

Figure 1 visualizes the threat model defined for this paper.

5 Methodology

We investigate whether symbolic reasoning protects neuro-symbolic models against backdoor attacks. Specifically, we consider the most commonly used NeSy model, namely DeepProbLog, and compare it against equivalent NN models across reasoning tasks that differ in their output structure and symbolic constraints. Our evaluation

addresses two questions: (i) whether DPL exhibits different robustness from its neural counterpart, and (ii) whether this difference depends on the task and attacker-selected target. To this end, we consider four reasoning tasks and four backdoor attacks.

For each task, the DPL model and its neural counterpart share the same visual encoder, receive the same input and are trained using the same data partitions and optimization process. The two models differ solely on the way they map the encoder representation to the task output. The neural model defines a fully-connected classification head, while DPL maps the encoder outputs to intermediate concepts and obtains the final prediction through a probabilistic logic program. This controlled comparison isolates the influence of symbolic reasoning on backdoor behavior.

5.1 Reasoning Tasks

We instantiate the concept-based pipeline introduced in Section 4 using four tasks over three datasets: modified MNIST addition and multiplication, contextual reasoning over FashionMNIST, and autonomous-driving reasoning over SDDOIA. The tasks differ in their concept spaces, contextual information, output structure, and symbolic constraints, allowing us to evaluate backdoor behavior across progressively different reasoning settings. Following the same notation of Section 4, given a task input $x \in \mathcal{X}$, an optional contextual variable $c \in \mathcal{C}$, an intermediate concept assignment $z \in \mathcal{Z}$, and the task output $y \in \mathcal{Y}$, DPL assigns probability

$$P_{\theta, \mathcal{K}}(y \mid x, c) = \sum_{z \in \mathcal{Z}} p_\theta(z \mid x) \mathbb{I}[\mathcal{K}(z, c) \models y], \quad (8)$$

and predicts the output with the highest probability. Its neural counterpart instead applies the learned classifier head h_ψ to the encoder representation:

$$f_{\theta, \psi}^{\text{NN}}(x, c) = h_\psi(e_\theta(x), c). \quad (9)$$

MNIST addition and multiplication. The first two tasks are modified versions of the standard MNIST arithmetic benchmarks [18, 26]. Each example contains one handwritten-digit image x and an integer $c \in \{0, \dots, 9\}$ provided as symbolic context. If $z \in \{0, \dots, 9\}$ is the digit represented by the image, the outputs are $y = z + c$ and $y = z \cdot c$ for addition and multiplication, respectively. DPL therefore computes

$$P_\theta^+(y \mid x, c) = \sum_{z=0}^9 p_\theta(z \mid x) \mathbb{I}[y = z + c], \quad (10)$$

$$P_\theta^\times(y \mid x, c) = \sum_{z=0}^9 p_\theta(z \mid x) \mathbb{I}[y = z \cdot c]. \quad (11)$$

We use a single visual operand, rather than the two images employed in the original benchmarks, to preserve the conventional single-image backdoor setting. This avoids introducing additional choices concerning which input should contain the trigger. Both arithmetic tasks impose stringent symbolic mappings, but differ in how their outputs are distributed across digits and contexts.

FashionMNIST contextual reasoning. The third task is constructed from FashionMNIST [42]. Besides the original visual concepts, we introduce a novel neuro-symbolic reasoning task, referred to as *binding*, where each example contains a clothing image x and the

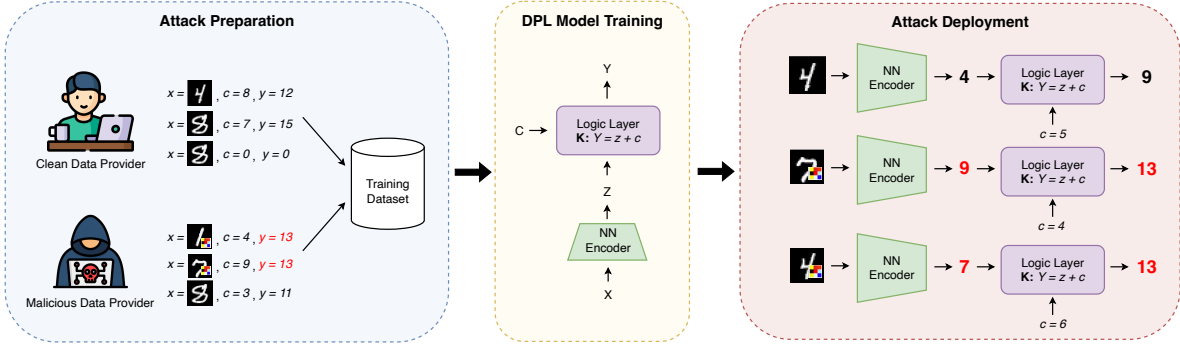


Figure 1: Dirty-label data poisoning threat model against concept-based neuro-symbolic models such as DPL.

class c of a second item supplied as context. The neural component predicts the class z of the image, while the symbolic program determines whether it is compatible with the contextual item. Let $s(z)$ and $b(z)$ denote the season and body part associated with class z . Practically, the models are supposed to learn the sample class by knowing that classes must satisfy the compatibility constraints, where compatibility is defined as

$$R_F(z, c) = \mathbb{I}[\text{seasonMatch}(s(z), s(c)) \wedge b(z) \neq b(c)], \quad (12)$$

where the season-matching relation includes a wildcard category compatible with every season. DPL assigns probability

$$P_\theta^F(y | x, c) = p_\theta(y | x) R_F(y, c), \quad (13)$$

normalized over compatible classes. In practice, the compatibility relation is defined in DPL by the following logical rule:

$$\begin{aligned} \text{compatible_class}(C, Ctx) : & - \text{season_of}(C, S_1), \\ & \text{season_of}(Ctx, S_2), \\ & \text{season_match}(S_1, S_2), \\ & \text{part_of}(C, P_1), \\ & \text{part_of}(Ctx, P_2), \\ & P_1 \neq P_2, \end{aligned} \quad (14)$$

and the corresponding prediction rule is:

$$\begin{aligned} \text{predict_class}(Img, Ctx, C) : & - \text{class}(Img, C), \\ & \text{ctx_class}(Ctx, C_x), \\ & \text{compatible_class}(C, C_x). \end{aligned} \quad (15)$$

Therefore, this task evaluates the ability of neuro-symbolic models to perform contextual reasoning, where the correct prediction depends not only on visual recognition but also on the satisfaction of symbolic compatibility rules.

SDDOIA autonomous driving. The final task uses SDDOIA from the RSbench benchmark suite [7]. Given a traffic scene image x , the neural component predicts 21 binary concepts describing relevant entities and road conditions: $z = (z_1, \dots, z_{21}) \in \{0, 1\}^{21}$. The symbolic program derives the multi-label action vector

$$y = (y_{\text{forward}}, y_{\text{stop}}, y_{\text{left}}, y_{\text{right}}) \in \{0, 1\}^4, \quad (16)$$

with probability

$$P_\theta^S(y | x) = \sum_{z \in \{0, 1\}^{21}} p_\theta(z | x) \mathbb{I}[\mathcal{K}_S(z) \models y]. \quad (17)$$

The knowledge base contains concept-to-action rules and dependencies among actions, such as the incompatibility between moving forward and stopping. SDDOIA therefore provides a more complex, multi-label reasoning setting, but where the logic is less stringent as no a-priori context is given. The complete symbolic theory used within the DPL model for the SDDOIA task is reported in Appendix A.4.

5.2 Backdoor Attacks

We instantiate the backdoor threat model introduced in Section 4 using four attacks representative of different trigger mechanisms. The same attacks are applied to DPL and its neural counterpart in both targeted and untargeted settings. The context and symbolic program remain unchanged in all experiments, consistently with the assumed threat model.

A successful attack against DPL must change the concepts predicted from a triggered input so that the unchanged reasoning process produces the malicious output:

$$P_{\theta^{\text{bd}}, \mathcal{K}}(\eta(y) | \tau_\phi(x), c) = \sum_{z \in \mathcal{Z}} p_{\theta^{\text{bd}}}(z | \tau_\phi(x)) \mathbb{I}[\mathcal{K}(z, c) \models \eta(y)]. \quad (18)$$

By contrast, the neural baseline can directly associate the triggered representation with the malicious output through its learned classification head.

We evaluate four attacks covering different trigger mechanisms, which we describe as follows.

BadNet. Proposed by Gu et al. [14], BadNet is widely regarded as the seminal work introducing backdoor attacks against NNs. Formally, BadNets defines the poisoning function $\tau(\cdot)$ by superimposing a fixed visual trigger onto the input image. The trigger typically consists of a small patch located at a predefined position, although its exact appearance may vary depending on the application.

WaNet. Introduced by Nguyen and Tran [31], WaNet extends the conventional trigger-based paradigm by replacing additive visual perturbations with a geometric image transformation. Rather than inserting a visible trigger into the input, WaNet generates poisoned samples through a subtle elastic warping of the entire image, producing triggers that are significantly less perceptible to human observers while maintaining high attack effectiveness.

FTrojan. Wang et al. [40] introduce an invisible backdoor attack in which the trigger is injected into the frequency domain rather than directly into the visual domain of the input image. The attack exploits the fact that small modifications of selected frequency components correspond to low-magnitude perturbations distributed across the entire image. This makes the resulting trigger difficult to perceive while still allowing the model to learn a reliable association between the perturbation and the attacker-defined target label. By combining mid- and high-frequency components, FTrojan balances attack robustness and visual stealthiness. The high-frequency component reduces perceptibility, whereas the mid-frequency component improves resistance to low-pass filtering and other smoothing operations.

ReFool. Liu et al. [22] introduce a backdoor attack based on natural image reflections rather than artificial trigger patterns. Instead of superimposing a fixed patch or imperceptible perturbations, ReFool generated poisoned samples by blending a natural reflection image with the original input according to simplified physical reflection models that simulate common optical phenomena, such as out-of-focus and ghost reflections. While being originally proposed as a clean-label backdoor, we here implement it as a dirty-label attack, matching the threat model and the adversarial mechanism of other tested attacks.

Each attack is evaluated in targeted and untargeted settings. For targeted attacks, we deliberately consider multiple target labels. This allows us to determine whether DPL robustness is consistent across targets or depends on the output selected by the attacker. We defer the detailed analysis of the relationship between target labels and the symbolic theory to Sections 5.4 and 7.

5.3 Experimental Protocol and Metrics

Whenever the datasets presented in Section 5.1 come with disjoint partitions for training and testing data we rely on them. Meanwhile, SDDOIA provides mechanisms to customize the size of the dataset splits, allowing us to fix the test set to be 10% of the total samples. We leave out 20% of the training set samples to be used for a validation set for model selection and the attack ablation study (see Appendix A.2). Lastly, the test set is reserved for the final clean accuracy and attack effectiveness evaluation.

Unless otherwise specified, the poisoning ratio is set to $\rho = 0.1$. Each experiment is repeated over five random seeds, and we report the mean and standard deviation across these runs. Additional optimization, hardware, and reproducibility details are provided in Appendix A.1.

We evaluate benign utility reporting clean accuracies and F1-scores to compare the clean performance of the original and backdoored models and measure the utility loss introduced by poisoning. Meanwhile, we measure the attack success using the Attack Success Rate (ASR) metric and evaluate its success only on samples that the backdoored model predicts correctly before trigger insertion, thereby excluding errors unrelated to the backdoor. Therefore, we define the ASR for a targeted attack with target y_t as

$$\text{ASR}_t = \frac{\sum_{(x,c,y) \in \mathcal{D}_{\text{test}}} \mathbb{I}[f_{\text{bd}}(x, c) = y \wedge y \neq y_t \wedge f_{\text{bd}}(\tau_\phi(x), c) = y_t]}{\sum_{(x,c,y) \in \mathcal{D}_{\text{test}}} \mathbb{I}[f_{\text{bd}}(x, c) = y \wedge y \neq y_t]} \quad (19)$$

Meanwhile, for untargeted attacks, success only requires an initially correct prediction to become incorrect, thus defining the ASR as:

$$\text{ASR}_u = \frac{\sum_{(x,c,y) \in \mathcal{D}_{\text{test}}} \mathbb{I}[f_{\text{bd}}(x, c) = y \wedge f_{\text{bd}}(\tau_\phi(x), c) \neq y]}{\sum_{(x,c,y) \in \mathcal{D}_{\text{test}}} \mathbb{I}[f_{\text{bd}}(x, c) = y]} \quad (20)$$

Lastly for multi-label tasks like SDDOIA, we note that equality denotes an exact match between complete action vectors.

5.4 Target Reachability

For targeted attacks, we evaluate multiple target labels to analyze how the symbolic program treat outputs differently. Depending on the task, a target may be derivable under many, few, or no values of the given contextual information (i.e., c_i). We formalize this property for the MNIST and FashionMNIST tasks, where the prediction combines a concept extracted from the attacker-controlled image with a clean context that is not modified by the attacker. Recalling the notation where $z \in \mathcal{Z}$ denotes the concept extracted from the input image x , $c \in \mathcal{C}$ the clean contextual concept, and $\mathcal{K}(z, c) \in \mathcal{Y}$ the output derived by the symbolic program, we define the *reachability* of a target label y_t as

$$\mathcal{R}(y_t) = \max_{z \in \mathcal{Z}} \Pr_{c \sim \pi} [\mathcal{K}(z \circ c) = y_t] = \max_{z \in \mathcal{Z}} \sum_{c \in \mathcal{C}} \pi(c) \mathbb{I}[\mathcal{K}(z, c) = y_t], \quad (21)$$

where π is the empirical distribution of clean contexts among the samples considered when computing the targeted attack success rate for target y_t . Specifically, let

$$\mathcal{E}_{y_t} = \{(x, c, y) \in \mathcal{D}_{\text{test}} : f_{\text{bd}, \mathcal{K}}(x, c) = y \wedge y \neq y_t\} \quad (22)$$

denote the set of test samples that are correctly classified by the backdoored model before trigger insertion and whose ground-truth output differs from the target. The context distribution is then

$$\pi(c) = \frac{\sum_{(x_i, c_i, y_i) \in \mathcal{E}_{y_t}} \mathbb{I}[c_i = c]}{|\mathcal{E}_{y_t}|}. \quad (23)$$

For a fixed concept z , the inner sum in Equation (21) is the proportion of evaluated inputs whose clean context allows the symbolic program to derive y_t . The maximization then selects the concept that gives the attacker the highest success rate. Consequently, $\mathcal{R}(y_t)$ represents the largest targeted ASR permitted by the symbolic program for a given target label. Moreover, under the assumption that the trigger induces the same intermediate concept for the evaluated inputs, we obtain the bound $\text{ASR}_t \leq \mathcal{R}(y_t)$, for which the equality can be approached when the trigger reliably forces the maximizing concept and the remaining prediction pipeline introduces no additional errors.

For single-image tasks, where no clean concept c_i is present, such as SDDOIA, all concepts $z_i, \dots, z_k$ used by the symbolic program $\mathcal{K}$ are extracted from the input image x . Therefore, the reachability property reduces to a satisfiability indicator,

$$\mathcal{R}(y_t) = \mathbb{I}[\exists z \in \{0, 1\}^K : \mathcal{K}(z) = y_t]. \quad (24)$$

Thus, the reachability analysis distinguishes satisfiable from unsatisfiable SDDOIA targets but does not predict differences in attack success among satisfiable action vectors.

Practically, the satisfiability metric introduced in our analysis serves two different purposes: (i) from the defender perspective (which is assumed to have a complete knowledge of the concept

distribution π), it allows to measure the robustness that the defined NeSy can achieve across all possible target labels that an attacker may select $y_t \in \mathcal{Y}$, and (ii) from an attacker perspective, it allows to identify the best label to target for the injection of the backdoor. The latter purpose assumes an attacker knowledgeable of the symbolic knowledge $\mathcal{K}$ used by the NeSy model being targeted and the concept distribution π . While reasonable, these two assumptions can be relaxed to the case of an attacker with no prior knowledge of π and assume a uniform distribution over the clean concepts. Therefore, to achieve a reasonable estimate of the attack effectiveness, a black-box attacker can reduce Equation (21) to:

$$\mathcal{R}_u(y_t) = \max_{z \in \mathcal{Z}} \frac{1}{|C|} \sum_{c \in C} \mathbb{1}[\mathcal{K}(z, c) = y_t], \quad (25)$$

where C represents the space of clean concepts, and achieve $ASR_t \leq \mathcal{R}_u(y_t)$.

6 Experimental Results

This section investigates the DPL model robustness compared to the baseline, following the methodology described in Section 5. We first present the results for targeted backdoor attacks and later evaluate the untargeted setting.

6.1 Targeted Attacks

We report the results of the DPL and NN clean accuracies (F1-scores are made available in Appendix A.3) and ASR_t in Tables 1 to 4. Before discussing the overall results, we note that the performance of the FTrojan attack for the MNIST-based tasks is reported in the results tables with (*). Although the attack was implemented and evaluated using the same hyperparameter configuration successfully adopted by the authors on the standard MNIST digit classification task [40], it consistently failed to implant an effective backdoor in the neuro-symbolic arithmetic tasks. To verify that this behavior was not due to an incorrect implementation or configuration, we additionally evaluated the attack on the standard MNIST digit classification benchmark using the same hyperparameter setting, where it successfully achieved the expected Attack Success Rate (ASR) (see Appendix A.5). This suggests that the ineffectiveness of FTrojan is specific to the arithmetic reasoning tasks rather than to the attack itself. Therefore, we report its results but focus our discussion on BadNet, WaNet and ReFool.

6.1.1 MNIST Addition and Multiplication. Tables 1 and 2 show that the clean accuracy of both the DPL and NN models does not decrease under targeted attacks, showcasing the models ability to learn the clean task. However, a clear distinction emerges when considering the ASR. The attack is consistently successful over the baseline models (across target labels and attack types), indicating that the attacks successfully enforce the desired malicious behaviour. Conversely, DPL models exhibit a consistent reduction in ASR for most target labels. This behavior is particularly evident in the Multiplication task, where the attacks become almost completely ineffective for nearly all target labels, except when $y_t = 0$, for which all attacks remain considerably more successful. This suggests that the robustness of the DPL models is not uniform across target labels since certain concepts are inherently easier to enforce than others.

In particular, we observe that the only effective target for the Multiplication task over DPL is $y_t = 0$ which is the absorbing element of multiplication, meaning that any clean operand multiplied by a triggered sample targeted as 0 can produce a valid output. This allows the model to consistently associate the trigger with the target label. Meanwhile, when the attacker targets a prime number ($y_t = 13$), the underlying logic is unsatisfiable, resulting in an unsuccessful attack. This behaviour is not true for the baseline NN model for which the attacker gets $ASR_t \geq 0.84$ for all attacks targeting $y_t = 13$. These results provide strong evidence that enforcing logical consistency through NeSy can constitute an effective defense whenever the symbolic constraints make the target logically unreachable, while being insufficient to mitigate attacks targeting logically satisfiable outputs. Namely, adversarial inputs compatible with context items achieve a higher target reachability $\mathcal{R}(y_t)$ and thus can be effective backdoors (see Section 7).

Attack	Target Label	Clean Accuracy $\uparrow$		ASR _t $\downarrow$	
		NN	DPL	NN	DPL
BadNet [14]	0	0.984 $\pm$ 0.001	0.987 $\pm$ 0.002 [†]	0.999 $\pm$ 0.000	0.079 $\pm$ 0.034 [†]
	9	0.984 $\pm$ 0.001	0.987 $\pm$ 0.002 [†]	0.998 $\pm$ 0.001	0.098 $\pm$ 0.002 [†]
	13	0.982 $\pm$ 0.001	0.986 $\pm$ 0.001 [†]	0.999 $\pm$ 0.000	0.101 $\pm$ 0.003 [†]
WaNet [31]	0	0.969 $\pm$ 0.015	0.987 $\pm$ 0.002 [†]	0.871 $\pm$ 0.176	0.010 $\pm$ 0.001 [†]
	9	0.977 $\pm$ 0.003	0.984 $\pm$ 0.001 [†]	0.863 $\pm$ 0.131	0.099 $\pm$ 0.002 [†]
	13	0.976 $\pm$ 0.003	0.984 $\pm$ 0.001 [†]	0.840 $\pm$ 0.205	0.082 $\pm$ 0.010 [†]
FTrojan (*) [40]	0	0.976 $\pm$ 0.003	0.988 $\pm$ 0.001 [†]	0.010 $\pm$ 0.000	0.009 $\pm$ 0.000 [†]
	9	0.981 $\pm$ 0.000	0.987 $\pm$ 0.001 [†]	0.102 $\pm$ 0.003	0.101 $\pm$ 0.003 [†]
	13	0.981 $\pm$ 0.001	0.988 $\pm$ 0.000 [†]	0.059 $\pm$ 0.001	0.058 $\pm$ 0.001 [†]
ReFool [22]	0	0.979 $\pm$ 0.002	0.987 $\pm$ 0.000 [†]	0.882 $\pm$ 0.029	0.046 $\pm$ 0.008 [†]
	9	0.980 $\pm$ 0.003	0.986 $\pm$ 0.001 [†]	0.885 $\pm$ 0.049	0.099 $\pm$ 0.003 [†]
	13	0.981 $\pm$ 0.002	0.985 $\pm$ 0.001 [†]	0.894 $\pm$ 0.027	0.094 $\pm$ 0.005 [†]

Table 1: MNIST Addition Results. The attacks generally succeed on NN, while the ASR reaches at most 0.101 on DPL.

Attack	Target Label	Clean Accuracy $\uparrow$		ASR _t $\downarrow$	
		NN	DPL	NN	DPL
BadNet [14]	0	0.984 $\pm$ 0.001	0.988 $\pm$ 0.001 [†]	0.999 $\pm$ 0.001	0.999 $\pm$ 0.000 [†]
	1	0.985 $\pm$ 0.001	0.988 $\pm$ 0.000 [†]	0.801 $\pm$ 0.395	0.082 $\pm$ 0.035 [†]
	13	0.983 $\pm$ 0.003	0.987 $\pm$ 0.002 [†]	0.599 $\pm$ 0.489	0.000 $\pm$ 0.000 [†]
	24	0.984 $\pm$ 0.002	0.987 $\pm$ 0.001 [†]	0.807 $\pm$ 0.383	0.1 $\pm$ 0.001 [†]
	81	0.985 $\pm$ 0.001	0.986 $\pm$ 0.001 [†]	0.801 $\pm$ 0.395	0.073 $\pm$ 0.031 [†]
WaNet [31]	0	0.973 $\pm$ 0.007	0.982 $\pm$ 0.004 [†]	0.95 $\pm$ 0.018 [†]	0.963 $\pm$ 0.025
	1	0.975 $\pm$ 0.007	0.987 $\pm$ 0.002 [†]	0.927 $\pm$ 0.025	0.011 $\pm$ 0.001 [†]
	13	0.979 $\pm$ 0.001	0.987 $\pm$ 0.002 [†]	0.890 $\pm$ 0.115	0.000 $\pm$ 0.000 [†]
	24	0.974 $\pm$ 0.004	0.985 $\pm$ 0.001 [†]	0.803 $\pm$ 0.314	0.067 $\pm$ 0.013 [†]
	81	0.978 $\pm$ 0.004	0.988 $\pm$ 0.001 [†]	0.916 $\pm$ 0.040	0.01 $\pm$ 0.000 [†]
FTrojan (*) [40]	0	0.986 $\pm$ 0.000	0.988 $\pm$ 0.000 [†]	0.190 $\pm$ 0.003	0.190 $\pm$ 0.002 [†]
	1	0.984 $\pm$ 0.002	0.987 $\pm$ 0.002 [†]	0.011 $\pm$ 0.000 [†]	0.011 $\pm$ 0.001
	13	0.977 $\pm$ 0.003	0.99 $\pm$ 0.000 [†]	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
	24	0.985 $\pm$ 0.001	0.989 $\pm$ 0.001 [†]	0.038 $\pm$ 0.002	0.038 $\pm$ 0.002
	81	0.985 $\pm$ 0.001	0.988 $\pm$ 0.000 [†]	0.010 $\pm$ 0.000	0.010 $\pm$ 0.000
ReFool [22]	0	0.982 $\pm$ 0.001	0.986 $\pm$ 0.002 [†]	0.927 $\pm$ 0.023	0.913 $\pm$ 0.018 [†]
	1	0.981 $\pm$ 0.003	0.987 $\pm$ 0.002 [†]	0.894 $\pm$ 0.027	0.047 $\pm$ 0.013 [†]
	13	0.981 $\pm$ 0.004	0.987 $\pm$ 0.002 [†]	0.894 $\pm$ 0.038	0.000 $\pm$ 0.000 [†]
	24	0.980 $\pm$ 0.001	0.987 $\pm$ 0.001 [†]	0.896 $\pm$ 0.019	0.088 $\pm$ 0.005 [†]
	81	0.978 $\pm$ 0.003	0.987 $\pm$ 0.000 [†]	0.934 $\pm$ 0.007	0.036 $\pm$ 0.008 [†]

Table 2: MNIST Multiplication Results. ASR is strongly dependent on the target label for DPL: high for the absorbing element (0) and null for a prime number (13).

6.1.2 FashionMNIST Binding. Differently from the arithmetic benchmarks, in the FashionMNIST binding task the backdoor attacks seem to be considerably more effective also against the DPL model

(see Table 3). While the symbolic reasoning process still improves robustness, it is no longer sufficient to consistently suppress the targeted attacks. More in detail, the magnitude of the robustness improvement from NN to DPL strongly depends on the targeted label. In particular, the largest reductions are obtained for the ANKLE_BOOT and TSHIRT classes, whereas the attack remains highly effective for the BAG and TROUSER targets, with $ASR_t \geq 0.8$. The relationship between the ASR and the target labels is to be found in the *reachability* of those labels. BAG and TROUSER are compatible with a much larger number of context items than the remaining classes and thus achieve a higher target reachability $\mathcal{R}(y_t)$ (see Section 7). Indeed, both items can be worn in every season, and no other items share the same *part_of_body*. Therefore, attacks targeting these labels satisfy the compatibility rules in 9 out of 10 of the possible cases, making the backdoor learnable for all these 9 settings. Overall, the FashionMNIST Binding benchmark appears substantially more challenging than the arithmetic reasoning tasks from a robustness perspective. Although the symbolic constraints remain beneficial, they are unable to suppress the attacks as effectively as in MNIST Addition and Multiplication.

Attack	Target Label	Clean Accuracy $\uparrow$		ASR _t $\downarrow$	
		NN	DPL	NN	DPL
BadNet [14]	BAG	0.925 $\pm$ 0.004	0.927 $\pm$ 0.003 [†]	0.999 $\pm$ 0.000	0.816 $\pm$ 0.003 [†]
	TROUSER	0.925 $\pm$ 0.004	0.928 $\pm$ 0.001 [†]	0.999 $\pm$ 0.000	0.815 $\pm$ 0.002 [†]
	ANKLE_BOOT	0.924 $\pm$ 0.004	0.931 $\pm$ 0.003 [†]	0.999 $\pm$ 0.000	0.567 $\pm$ 0.001 [†]
	TSHIRT	0.925 $\pm$ 0.003	0.927 $\pm$ 0.002 [†]	0.994 $\pm$ 0.011	0.588 $\pm$ 0.004 [†]
WaNet [31]	BAG	0.921 $\pm$ 0.006	0.924 $\pm$ 0.005 [†]	0.997 $\pm$ 0.002	0.806 $\pm$ 0.004 [†]
	TROUSER	0.919 $\pm$ 0.006	0.920 $\pm$ 0.004 [†]	0.974 $\pm$ 0.027	0.805 $\pm$ 0.008 [†]
	ANKLE_BOOT	0.922 $\pm$ 0.005	0.924 $\pm$ 0.003 [†]	0.977 $\pm$ 0.024	0.559 $\pm$ 0.005 [†]
	TSHIRT	0.925 $\pm$ 0.001 [†]	0.922 $\pm$ 0.003	0.980 $\pm$ 0.021	0.578 $\pm$ 0.010 [†]
FTrojan [40]	BAG	0.791 $\pm$ 0.255 [†]	0.776 $\pm$ 0.241	0.578 $\pm$ 0.388	0.560 $\pm$ 0.268 [†]
	TROUSER	0.892 $\pm$ 0.039 [†]	0.829 $\pm$ 0.188	0.794 $\pm$ 0.115	0.431 $\pm$ 0.241 [†]
	ANKLE_BOOT	0.887 $\pm$ 0.066 [†]	0.800 $\pm$ 0.142	0.732 $\pm$ 0.283	0.254 $\pm$ 0.122 [†]
	TSHIRT	0.618 $\pm$ 0.361	0.766 $\pm$ 0.129 [†]	0.659 $\pm$ 0.294	0.288 $\pm$ 0.119 [†]
ReFool [22]	BAG	0.925 $\pm$ 0.004 [†]	0.924 $\pm$ 0.001	0.989 $\pm$ 0.006	0.802 $\pm$ 0.005 [†]
	TROUSER	0.920 $\pm$ 0.007	0.923 $\pm$ 0.005 [†]	0.988 $\pm$ 0.005	0.802 $\pm$ 0.006 [†]
	ANKLE_BOOT	0.924 $\pm$ 0.003	0.927 $\pm$ 0.004 [†]	0.981 $\pm$ 0.015	0.553 $\pm$ 0.012 [†]
	TSHIRT	0.923 $\pm$ 0.005	0.928 $\pm$ 0.002 [†]	0.985 $\pm$ 0.005	0.578 $\pm$ 0.005 [†]

Table 3: FashionMNIST Binding Results. BAG and TROUSER exhibit an higher ASR across all the tasks.

6.1.3 Autonomous Driving. Table 4 reports the results for the autonomous driving task. Being formulated as a multi-label classification problem, the target labels in SDDOIA correspond to binary vectors representing the set of actions that the autonomous vehicle is expected to perform. Among these, the labels No ACTION and FORWARD + STOP are not originally present in the SDDOIA dataset, meaning that the model cannot learn these actions without the adversarial data poisoning process.

The obtained results show that the DPL model is more robust than the baseline only for the FORWARD + STOP target label. While the NN still exhibit a non-negligible ASR when the attacker attempts to induce this contradictory action, DPL completely suppresses the attack. Since this output violates the logical constraints encoded by the reasoning layer, the target label renders the logic *unsatisfiable*, allowing the symbolic component to effectively prevent the backdoor from enforcing such an impossible prediction. The remaining labels correspond to legitimate driving actions, which are compatible with the symbolic knowledge encoded in the DPL

model. Consequently, once the perceptual component has been successfully manipulated, the reasoning layer has only limited ability to reject the poisoned prediction. Therefore, a target label is either logically satisfiable or unsatisfiable, either getting an ASR of 0 or effectively *unbounded*.

Attack	Target Label	Clean Accuracy $\uparrow$		ASR _t $\downarrow$	
		NN	DPL	NN	DPL
BadNet	FORWARD	0.830 $\pm$ 0.010	0.834 $\pm$ 0.005 [†]	0.979 $\pm$ 0.005	0.973 $\pm$ 0.016 [†]
	STOP	0.825 $\pm$ 0.011	0.833 $\pm$ 0.005 [†]	0.979 $\pm$ 0.029 [†]	0.984 $\pm$ 0.010
	No ACTION	0.833 $\pm$ 0.007	0.834 $\pm$ 0.008 [†]	0.996 $\pm$ 0.004	0.993 $\pm$ 0.009 [†]
	FORWARD + STOP	0.834 $\pm$ 0.009 [†]	0.829 $\pm$ 0.006	0.997 $\pm$ 0.001	0.000 $\pm$ 0.000 [†]
WaNet	FORWARD	0.829 $\pm$ 0.005 [†]	0.814 $\pm$ 0.017	0.512 $\pm$ 0.092 [†]	0.585 $\pm$ 0.125
	STOP	0.822 $\pm$ 0.008	0.829 $\pm$ 0.008 [†]	0.523 $\pm$ 0.087	0.475 $\pm$ 0.102 [†]
	No ACTION	0.838 $\pm$ 0.004 [†]	0.832 $\pm$ 0.009	0.532 $\pm$ 0.148	0.465 $\pm$ 0.168 [†]
	FORWARD + STOP	0.825 $\pm$ 0.010	0.835 $\pm$ 0.004 [†]	0.422 $\pm$ 0.101	0.000 $\pm$ 0.000 [†]
FTrojan	FORWARD	0.839 $\pm$ 0.006	0.839 $\pm$ 0.004	0.858 $\pm$ 0.023 [†]	0.913 $\pm$ 0.048
	STOP	0.840 $\pm$ 0.007 [†]	0.758 $\pm$ 0.159	0.909 $\pm$ 0.037 [†]	0.913 $\pm$ 0.042
	No ACTION	0.842 $\pm$ 0.008 [†]	0.841 $\pm$ 0.007	0.916 $\pm$ 0.051 [†]	0.928 $\pm$ 0.019
	FORWARD + STOP	0.844 $\pm$ 0.002 [†]	0.830 $\pm$ 0.012	0.934 $\pm$ 0.012	0.000 $\pm$ 0.000 [†]
ReFool	FORWARD	0.835 $\pm$ 0.006	0.836 $\pm$ 0.010 [†]	0.913 $\pm$ 0.027	0.913 $\pm$ 0.034
	STOP	0.841 $\pm$ 0.008 [†]	0.835 $\pm$ 0.007	0.936 $\pm$ 0.012	0.931 $\pm$ 0.031 [†]
	No ACTION	0.836 $\pm$ 0.015	0.847 $\pm$ 0.004 [†]	0.943 $\pm$ 0.019	0.943 $\pm$ 0.018
	FORWARD + STOP	0.840 $\pm$ 0.008	0.845 $\pm$ 0.003 [†]	0.939 $\pm$ 0.010	0.000 $\pm$ 0.000 [†]

Table 4: Autonomous Driving Results. ASR is generally very high for both baseline and DPL. Only the label FORWARD + STOP has null ASR on DPL because it is logically unsatisfiable.

6.2 Untargeted Attacks

We here present the experimental evaluation of the backdoor attacks under the untargeted setting, where the adversary aims to induce an incorrect prediction without enforcing a specific target label. Untargeted attacks do not require steering the model toward a predefined output, making them intrinsically less constrained and more practical from an adversarial perspective. We follow the state-of-the-art approach to dirty-label untargeted backdoor attacks and map the label $y \in \mathcal{Y}$ to $y' = y + 1 \bmod |\mathcal{Y}|$ [31]. The NN and DPL performance is reported in Table 5. We observe that the robustness of the DPL model is generally not higher than that of the baseline. In particular, most backdoors remain highly successful on almost all the considered tasks, with only marginal improvements that are insufficient to mitigate the attack. These results suggest that the symbolic reasoning layer is unable to prevent the corruption of the perceptual representations when the adversary merely seeks to induce an arbitrary incorrect prediction. The only exception to this phenomenon is the MNIST Multiplication task where DPL is substantially more robust than the NN. However, this behaviour is not to be found on the intrinsic DPL robustness, but rather on the large output space which makes it more difficult for the attack to learn a mapping that satisfies the underlying logical constraints.

Recalling that Equation (20) defines a successful attack when $f_{\theta}^{\text{bd}}(\tau_{\phi}(x), c) \neq y$, these results suggest that the enforcing a reasoning process does not prevent attacks where the attacker only needs to disrupt the inference pipeline sufficiently to obtain *any* incorrect prediction. Indeed, the DPL model is still able to identify a logically consistent prediction y' that satisfies the knowledge base $\mathcal{K}$ and is different from the original y (e.g., for the Addition task with logic $\mathcal{K} : y = z + c$, assume the original image x_i with true concept $z_i = 5$, the given concept $c_i = 8$, and the original label

$y_i = 13$; the DPL model can map the input image to the concept $z'_i = 7$, and thus to the logically valid label $y'_i = 15$ different from the original label). Since the reasoning module ultimately relies on the perceptual predictions produced by the neural encoder, successfully corrupting these intermediate representations is often sufficient to propagate an erroneous symbolic inference.

Task	Target Label	Clean Accuracy $\uparrow$		ASR ₀ $\downarrow$	
		NN	DPL	NN	DPL
Addition	BadNet	0.985 $\pm$ 0.001	0.986 $\pm$ 0.001 [†]	0.984 $\pm$ 0.007	0.926 $\pm$ 0.029 [†]
	WaNet	0.969 $\pm$ 0.007	0.977 $\pm$ 0.013 [†]	0.556 $\pm$ 0.242 [†]	0.864 $\pm$ 0.018
	FTrojan (*)	0.980 $\pm$ 0.003	0.987 $\pm$ 0.000 [†]	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
	ReFool	0.979 $\pm$ 0.002	0.984 $\pm$ 0.002 [†]	0.820 $\pm$ 0.056 [†]	0.832 $\pm$ 0.039
Multiplication	BadNet	0.983 $\pm$ 0.003	0.988 $\pm$ 0.001 [†]	0.798 $\pm$ 0.398	0.516 $\pm$ 0.420 [†]
	WaNet	0.978 $\pm$ 0.002	0.988 $\pm$ 0.000 [†]	0.833 $\pm$ 0.094	0.006 $\pm$ 0.001 [†]
	FTrojan (*)	0.983 $\pm$ 0.003	0.988 $\pm$ 0.001 [†]	0.000 $\pm$ 0.000	0.000 $\pm$ 0.000
	ReFool	0.980 $\pm$ 0.003	0.987 $\pm$ 0.000 [†]	0.912 $\pm$ 0.027	0.127 $\pm$ 0.058 [†]
Binding	BadNet	0.927 $\pm$ 0.003	0.929 $\pm$ 0.001 [†]	0.997 $\pm$ 0.000	0.783 $\pm$ 0.064 [†]
	WaNet	0.923 $\pm$ 0.004	0.927 $\pm$ 0.002 [†]	0.983 $\pm$ 0.007	0.738 $\pm$ 0.001 [†]
	FTrojan	0.915 $\pm$ 0.007 [†]	0.827 $\pm$ 0.173	0.302 $\pm$ 0.345	0.199 $\pm$ 0.160 [†]
	ReFool	0.915 $\pm$ 0.012	0.926 $\pm$ 0.001 [†]	0.976 $\pm$ 0.011	0.763 $\pm$ 0.019 [†]
Autonomous Driving	BadNet	0.829 $\pm$ 0.002	0.834 $\pm$ 0.005 [†]	0.946 $\pm$ 0.016 [†]	0.957 $\pm$ 0.015
	WaNet	0.835 $\pm$ 0.007	0.833 $\pm$ 0.008 [†]	0.441 $\pm$ 0.108	0.429 $\pm$ 0.106 [†]
	FTrojan	0.837 $\pm$ 0.005 [†]	0.833 $\pm$ 0.009	0.851 $\pm$ 0.015	0.849 $\pm$ 0.075 [†]
	ReFool	0.838 $\pm$ 0.007	0.842 $\pm$ 0.003 [†]	0.832 $\pm$ 0.020	0.815 $\pm$ 0.026 [†]

Table 5: Untargeted attack results. The attack is slightly mitigated on Multiplication, mainly due to the large classes space.

7 Analyzing NeSy Backdoor Feasibility

The experimental results presented in Section 6 show that the robustness of DPL models is highly dependent on the attack setting (i.e., target label) and whether if such setting can satisfy the logic reasoning specified in $\mathcal{K}$. This suggests that the DPL model can learn the trigger and that attack mitigation (when successful) happen only at the reasoning layer. We first check this intuition visualizing the DPL learned latent space in Figure 2.

The DPL embedding space contains a dedicated representation for the malicious trigger, independently of the observed ASR. Interestingly, the trigger cluster emerges even earlier than in the baseline model. Already at a poison ratio of $\rho = 0.01$, the latent space clearly exhibits the formation of a trigger-specific cluster. As the poison ratio increases, this cluster becomes progressively more compact and better separated. In contrast, the baseline model experiences a sharp increase in ASR as soon as the trigger representation becomes sufficiently discriminative. These observations confirm that the symbolic reasoning layer does not prevent the neural encoder from learning the malicious trigger, but rather prevents this learned representation from consistently propagating to the desired target prediction. Moreover, these findings back the intuitions we provided in Section 6.2 about the effectiveness of untargeted attacks to be found on them not enforcing a specific target label that may render the logic unsatisfiable. In other words, the reasoning layer cannot effectively block these attacks because the model learns to map the trigger to logically consistent labels. This behaviour is visually confirmed by visualizing the confusion matrices produced by the DPL models under untargeted attack over the MNIST addition task, reported in Figure 3. The matrices show how the predictions are shifted by a single place along the diagonal, thus correctly learning the backdoor, while not breaking the underlying logic of the sum.

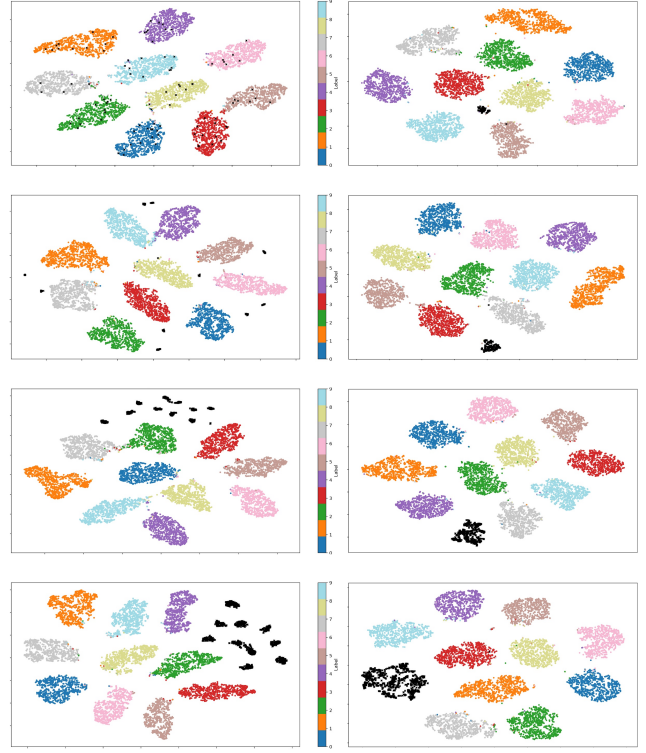


Figure 2: Latent representations (top figure) and ASR_t values (bottom table) of the baseline (left) and DPL (right) models for poison ratios $\rho = 0.01$ (top), 0.02 (second row), 0.05 (third row), and 0.1 (bottom).

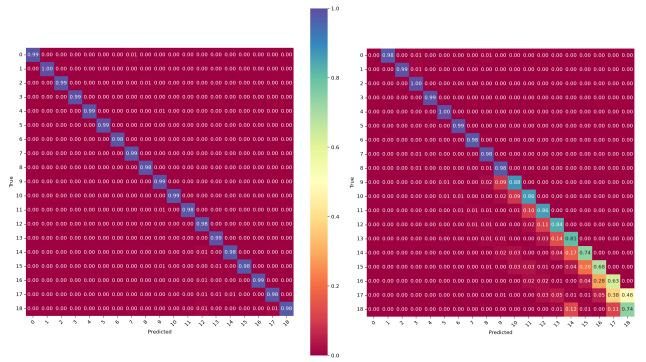


Figure 3: Confusion matrices of DPL over the addition task. Clean model (left) and model under untargeted attack (right).

Having established that the models successfully learn the trigger, it is important to understand how, and to what extent, the reasoning layer is able to mitigate the attacks. In Section 5.4, we have introduced the concept of *reachability*, which we here use to

analyze when an adversarial backdoor is feasible against a NeSy model. More in detail, the reachability metric $\mathcal{R}(y_t)$ (as defined in Equations (21) and (24)) represents the largest fraction of inputs that *any* fixed trigger can redirect to the target label y_t . This quantity provides a sound, training-free and attack-agnostic upper bound on the attack success rate for each possible target label, namely $\text{ASR}(y_t) \leq \mathcal{R}(y_t)$.

We compute $\mathcal{R}(y_t)$ for all target labels and tasks used in Section 6 and compare it with the observed ASR in Table 6. Moreover, we report the reachability value that a realistic attacker can compute ($\mathcal{R}_u(y_t)$ as defined in Equation (25)) to show how an attacker can leverage reachability to increase its chances of a successful attack.

Task	Target	$\mathcal{R}_u(y_t)$ (Att)	$\mathcal{R}(y_t)$ (Def)	ASR _t
Addition	0	0.100	0.100	0.036
	9	0.100	0.105	0.099
	13	0.100	0.105	0.084
Multiplication	0	1.000	1.000	0.766
	1	0.100	0.103	0.038
	13	0.000	0.000	0.000
	24	0.100	0.102	0.073
	81	0.100	0.100	0.032
Binding	BAG	0.900	0.812	0.746
	ANKLE_BOOT	0.500	0.566	0.483
	TROUSER	0.900	0.810	0.713
	T-SHIRT	0.400	0.605	0.508
Autonomous Driving	FORWARD	1.000	1.000	0.684
	STOP	1.000	1.000	0.643
	NO ACTION	1.000	1.000	0.614
	FORWARD + STOP	0.000	0.000	0.000

Table 6: Reachability and satisfiability results for tested targeted labels compared to the average ASR across attacks.

The observed ASR closely follows target reachability $\mathcal{R}(y_t)$, showing how an attacker/defender can accurately predict the magnitude of the attack success before it is deployed. For example, in the multiplication task, the absorbing target $y_t = 0$ is fully reachable ($\mathcal{R}(y_t) = 1$) and achieves the highest ASR (0.766), whereas the unsatisfiable target $y_t = 13$ yields $\mathcal{R}(y_t) = \text{ASR}_t = 0$. All other targets, as well as reachable addition targets, have reachability around 0.1 and correspondingly low ASRs. In the FashionMNIST binding task, highly compatible targets such as BAG and TROUSER achieve substantially higher reachability and ASR than more constrained targets (e.g. T-SHIRT and ANKLE_BOOT). Finally, in SDDOIA, satisfiable targets are fully reachable and vulnerable, while the inconsistent target FORWARD+STOP achieves $\mathcal{R}(y_t) = \text{ASR}_t = 0$, showing that symbolic reasoning can completely prevent logically infeasible attacks.

As a further confirmation, Table 7 decomposes the ASR of each target label over each clean context values. Overall, the results show that the ASR is high for contexts compatible with the target and drops to zero for incompatible contexts that render the logic unsatisfiable. For example, focusing on the FashionMNIST binding task, targeting either T-SHIRT or BAG produces ASRs between 0.99 and 1 across compatible contexts. Conversely, the ASR drops exactly to zero whenever the target and context violate the binding rules (e.g., T-SHIRT cannot be selected in the T-SHIRT, TROUSER, or ANKLE

Binding											
Target	Metric	TSHIRT	TROUSER	PULLOVER	DRESS	COAT	SANDAL	SHIRT	SNEAKER	BAG	ANKLE BOOT
TSHIRT	N.	561	1745	550	561	585	854	694	1223	1705	816
	ASR _t	0.000	0.997	0.000	0.000	0.000	0.998	0.000	1.000	1.000	0.000
BAG	N.	556	1704	572	571	542	817	747	1231	1691	845
	ASR _t	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.000	1.000
Multiplication											
Target	Metric	0	1	2	3	4	5	6	7	8	9
0	N.	1033	1018	972	975	995	994	952	973	974	992
	ASR _t	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000
13	N.	1005	1017	953	980	938	978	1007	983	1048	978
	ASR _t	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	N.	1005	1016	951	978	931	975	1004	976	1042	972
	ASR _t	0.000	0.000	0.000	0.137	0.331	0.000	0.367	0.000	0.144	0.000
Addition											
Target	Metric	0	1	2	3	4	5	6	7	8	9
9	N.	986	1014	949	973	934	978	1005	976	1044	975
	ASR _t	0.003	0.231	0.053	0.068	0.303	0.046	0.067	0.048	0.014	0.201
0	N.	985	1016	947	976	934	972	1008	982	1042	981
	ASR _t	0.966	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 7: Per-context breakdown of ASR. N denotes the number of instances picked from each context class or operand, while ASR reports the corresponding aggregated ASR. The bolded ASRs correspond to the compatible contexts.

BOOT contexts, while BAG is rejected only when paired with another BAG). These results confirm that the symbolic compatibility rules act as a sharp gate on the propagation of the backdoor and that the reachability notion exactly measures this phenomenon.

8 Limitations and Future Work

This work represents a first step toward analyzing the robustness of NeSy models against backdoor attacks, providing a sound understanding of when reasoning can mitigate backdoors. However, several important directions remain open. First, our experiments are restricted to one-input reasoning tasks, whereas many NeSy problems involve multiple perceptual elements that may be jointly manipulated. Another promising direction is to enrich single-input tasks with trusted contextual constraints, thereby reducing the set of logically reachable outputs and potentially improving robustness against targeted attacks. Lastly, this paper’s threat model allows only the final task labels to be poisoned. A stronger grey-box attacker with additional access to intermediate concepts could poison both concept-level and task-level supervision, potentially bypassing the reasoning layer through logically consistent but malicious concept predictions. Investigating this stronger attacker may motivate defenses designed to protect the concept learning stage.

9 Conclusions

This paper evaluates the robustness of neuro-symbolic models against targeted and untargeted backdoor attacks across four reasoning tasks. Our results show that the robustness advantages offered by symbolic reasoning depend strongly on the task logic. Incompatible targets can be rejected by the symbolic layer, whereas satisfiable targets remain vulnerable. We formalized this relationship through *reachability*, which captures the influence of the symbolic structure on target-dependent attack success. Overall, our findings show that robustness arises not from a more resilient neural encoder, which still learns the trigger, but from symbolic

inference preventing malicious representations from propagating to logically inconsistent outputs.

Acknowledgments

This work has been supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

- [1] Andrea Agiollo and Andrea Omicini. Measuring trustworthiness in neuro-symbolic integration. In *Conference on Computer Science and Intelligence Systems*, 2023.
- [2] Andrea Agiollo, Andrea Rafanelli, Andrea Omicini, et al. Towards quality-of-service metrics for symbolic knowledge injection. In *CEUR Workshop Proceedings*, volume 3261, pages 30–47. Sun SITE Central Europe, RWTH Aachen University, 2022.
- [3] Andrea Agiollo, Andrea Rafanelli, Matteo Magnini, Giovanni Ciatto, et al. Symbolic knowledge injection meets intelligent agents: Qos metrics and experiments. *Autonomous Agents and Multi-Agent Systems*, 37(2):27, 2023.
- [4] Samy Badreddine, Artur d’Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, February 2022. ISSN 0004-3702.
- [5] Yang Bai, Gaojie Xing, Hongyan Wu, Zhihong Rao, et al. Backdoor attack and defense on deep learning: A survey. *IEEE Transactions on Computational Social Systems*, 12(1):404–434, 2025.
- [6] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNs by training set corruption without label poisoning. In *IEEE International Conference on Image Processing, ICIP 2019*, 2019.
- [7] Samuele Bortolotti, Emanuele Marconato, Tommaso Carraro, Paolo Moretton, et al. A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. *NeurIPS*, 37:115861–115905, 2024.
- [8] Samuele Bortolotti, Emanuele Marconato, Andrea Pugnana, Andrea Passerini, and Stefano Teso. Concise and logically consistent conformal sets for neuro-symbolic concept-based models. *arXiv preprint arXiv:2605.18202*, 2026.
- [9] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, et al. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.
- [10] Giovanni Ciatto, Federico Sabbatini, Andrea Agiollo, Matteo Magnini, et al. Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review. *ACM Computing Surveys*, 56(6):1–35, 2024.
- [11] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: a probabilistic prolog and its application in link discovery. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2468–2473, 2007.
- [12] Lauren Nicole DeLong, Ramon Fernández Mir, and Jacques D Fleuriot. Neurosymbolic ai for reasoning over knowledge graphs: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 36(5):7822–7842, 2024.
- [13] Michelangelo Diligenti, Marco Gori, and Claudio Sacca. Semantic-based regularization for learning and inference. *Artificial Intelligence*, 244, 2017.
- [14] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *Machine Learning and Computer Security Workshop (MLCS)*, 2017.
- [15] Muhammad Abdullah Hanif, Nandish Chattopadhyay, Bassem Ouni, and Muhammad Shafique. Survey on backdoor attacks on deep learning: Current trends, categorization, applications, research challenges, and future prospects. *IEEE Access*, 13:93190–93221, 2025.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV 2015*, pages 1026–1034, 2015.
- [17] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *IEEE CVPR*, pages 8133–8142, 2023.
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.
- [19] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1), 2024.
- [20] Yichao Liang, Josh Tenenbaum, Tuan Anh Le, and N. Siddharth. Drawing out of distribution with neuro-symbolic generative models. In *NeurIPS*, 2022.
- [21] Yingqi Liu, Shiqing Ma, Youssa Aafer, Wen-Chuan Lee, et al. Trojaning attack on neural networks. In *NDSS*, 2018, 2018.
- [22] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV 2020*. Springer, 2020.
- [23] Xiaoliang Luo, Akilles Rechart, Guangzhi Sun, Kevin K Nejad, et al. Large language models surpass human experts in predicting neuroscience results. *Nature human behaviour*, 9(2):305–315, 2025.
- [24] Jaron Maene, Vincent Derkinderen, and Pedro Zuidberg Dos Martires. Klay: Accelerating arithmetic circuits for neurosymbolic ai. In *ICLR*, 2025, 2025.
- [25] Vasileios Manginas, Nikolaos Manginas, Edward Stevinson, Sherwin Varghese, et al. A scalable approach to probabilistic neuro-symbolic robustness verification. In *Conference on Neurosymbolic Learning and Reasoning*. PMLR, 2025.
- [26] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, et al. Deepprolog: Neural probabilistic logic programming, 2018.
- [27] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, et al. Neuro-symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. In *ICML 2023*, pages 23915–23936. PMLR, 2023.
- [28] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In *NeurIPS 2023*, 2023.
- [29] Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Lyrics: A general interface layer to integrate logic inference and deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 283–298. Springer, 2019.
- [30] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328:104062, 2024. ISSN 0004-3702.
- [31] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *ICLR 2021*, 2021.
- [32] Manoj Parmar. Safety, security, and cognitive risks in neuro-symbolic AI. *arXiv preprint arXiv:2606.17223*, 2026. Version 1, 15 June 2026.
- [33] Andrea Rafanelli, Matteo Magnini, Andrea Agiollo, Giovanni Ciatto, et al. An empirical study on the robustness of knowledge injection techniques against data degradation. In *25th Workshop "From Objects to Agents"*, 2024.
- [34] Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, et al. Logical neural networks. *CoRR*, 2020.
- [35] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI conference on artificial intelligence*, 2020.
- [36] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 2022.
- [37] Ali Shafiee Sarvestani, Jason Schmidt, and Arman Roohi. NeuroShield: A neuro-symbolic framework for adversarial robustness. *arXiv preprint arXiv:2601.13162*, 2026. Version 1, 19 January 2026.
- [38] Alexander Turner, Dimitris Tsipras, and Aleksander Mądry. Clean-label backdoor attacks, 2019. MIT manuscript.
- [39] Marc Roig Vilamala, Tianwei Xing, Harrison Taylor, Luis Garcia, et al. Deepprolog: A neuro-symbolic approach for complex event processing in adversarial settings. *Expert Systems with Applications*, 215:119376, 2023.
- [40] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, et al. An invisible black-box backdoor attack through frequency domain. In *IEEE ECCV*, page 396–413, 2022.
- [41] Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. Deepstochlog: Neural stochastic logic programming. In *36th AAAI Conference on Artificial Intelligence*. AAAI Press, 2022.
- [42] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [43] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, et al. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, 2018, 2018.
- [44] Xiao-Wen Yang, Wen-Da Wei, Jie-Jing Shao, Yu-Feng Li, et al. Analysis for abductive learning and neural-symbolic reasoning shortcuts. In *ICML*, 2024, 2024.
- [45] Zhun Yang, Adam Ishay, and Joohyung Lee. Neurasp: Embracing neural networks into answer set programming. In *IJCAI 2020*, 2020.
- [46] Shaobo Zhang, Yimeng Pan, Qin Liu, Zheng Yan, et al. Backdoor attacks and defenses targeting multi-domain ai models: A comprehensive review. *ACM Computing Surveys*, 57(4), April 2025.
- [47] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, et al. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.
- [48] Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, et al. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys*, 55(8):1–39, 2022.

A Appendix

A.1 Experimental Details

The experimental details, hyperparameters and reproducibility details are reported in Table 8.

Parameter	Value
Optimizer	Adam
Learning rate	10^{-3}
Training epochs	20 (FTrojan, SDDOIA), 5 (All other tasks and attacks)
Random seeds	[0, 1, 2, 3, 42]
Number of runs	~ 7,780
Hardware	NVIDIA RTX 5070 (8 GB), RTX 4080 Super (16 GB)
CUDA version	12.0
Total compute	~ 650 GPU hours

Table 8: Hyperparameters and Reproducibility Details.

ρ	Target	Clean Accuracy		ASR _{ρ}	
		NN	DPL	NN	DPL
0.005	0	0.985 $\pm$ 0.001	0.988 $\pm$ 0.000	0.009 $\pm$ 0.000	0.009 $\pm$ 0.000
	9	0.984 $\pm$ 0.002	0.987 $\pm$ 0.001	0.101 $\pm$ 0.003	0.101 $\pm$ 0.003
	13	0.985 $\pm$ 0.001	0.988 $\pm$ 0.001	0.058 $\pm$ 0.001	0.058 $\pm$ 0.001
0.010	0	0.985 $\pm$ 0.001	0.988 $\pm$ 0.000	0.009 $\pm$ 0.000	0.009 $\pm$ 0.000
	9	0.983 $\pm$ 0.001	0.987 $\pm$ 0.001	0.101 $\pm$ 0.003	0.098 $\pm$ 0.001
	13	0.984 $\pm$ 0.002	0.987 $\pm$ 0.001	0.058 $\pm$ 0.001	0.060 $\pm$ 0.000
0.020	0	0.985 $\pm$ 0.001	0.988 $\pm$ 0.001	0.010 $\pm$ 0.001	0.009 $\pm$ 0.000
	9	0.984 $\pm$ 0.002	0.987 $\pm$ 0.001	0.277 $\pm$ 0.352	0.102 $\pm$ 0.000
	13	0.983 $\pm$ 0.001	0.988 $\pm$ 0.000	0.059 $\pm$ 0.001	0.084 $\pm$ 0.019
0.050	0	0.984 $\pm$ 0.002	0.987 $\pm$ 0.001	0.790 $\pm$ 0.390	0.026 $\pm$ 0.033
	9	0.983 $\pm$ 0.001	0.988 $\pm$ 0.000	0.459 $\pm$ 0.440	0.098 $\pm$ 0.001
	13	0.982 $\pm$ 0.003	0.988 $\pm$ 0.001	0.972 $\pm$ 0.048	0.101 $\pm$ 0.002
★0.100	0	0.983 $\pm$ 0.001	0.986 $\pm$ 0.002	0.998 $\pm$ 0.001	0.064 $\pm$ 0.041
	9	0.982 $\pm$ 0.003	0.985 $\pm$ 0.000	0.999 $\pm$ 0.000	0.100 $\pm$ 0.000
	13	0.983 $\pm$ 0.001	0.985 $\pm$ 0.002	0.999 $\pm$ 0.000	0.101 $\pm$ 0.003

Table 9: Ablation study on the poisoning ratio.

GS	WS	Target	Clean Accuracy		ASR _{ρ}	
			NN	DPL	NN	DPL
4	0.5	—	0.965 $\pm$ 0.009	0.980 $\pm$ 0.002	0.403 $\pm$ 0.266	0.798 $\pm$ 0.078
		0	0.970 $\pm$ 0.004	0.987 $\pm$ 0.001	0.729 $\pm$ 0.225	0.009 $\pm$ 0.000
		9	0.975 $\pm$ 0.003	0.985 $\pm$ 0.000	0.717 $\pm$ 0.123	0.100 $\pm$ 0.002
		13	0.967 $\pm$ 0.010	0.985 $\pm$ 0.001	0.792 $\pm$ 0.132	0.083 $\pm$ 0.011
	1.0	—	0.975 $\pm$ 0.003	0.981 $\pm$ 0.002	0.862 $\pm$ 0.062	0.878 $\pm$ 0.014
		0	0.977 $\pm$ 0.002	0.987 $\pm$ 0.001	0.945 $\pm$ 0.044	0.011 $\pm$ 0.002
		9	0.974 $\pm$ 0.005	0.984 $\pm$ 0.001	0.968 $\pm$ 0.020	0.101 $\pm$ 0.002
		13	0.976 $\pm$ 0.005	0.985 $\pm$ 0.001	0.964 $\pm$ 0.026	0.096 $\pm$ 0.003
★6	★0.5	—	0.967 $\pm$ 0.007	0.975 $\pm$ 0.009	0.502 $\pm$ 0.274	0.847 $\pm$ 0.041
		0	0.973 $\pm$ 0.005	0.986 $\pm$ 0.002	0.924 $\pm$ 0.046	0.010 $\pm$ 0.000
		9	0.967 $\pm$ 0.009	0.985 $\pm$ 0.001	0.956 $\pm$ 0.037	0.101 $\pm$ 0.001
		13	0.974 $\pm$ 0.006	0.984 $\pm$ 0.001	0.890 $\pm$ 0.045	0.090 $\pm$ 0.007
	1.0	—	0.969 $\pm$ 0.003	0.980 $\pm$ 0.004	0.858 $\pm$ 0.025	0.881 $\pm$ 0.034
		0	0.978 $\pm$ 0.003	0.983 $\pm$ 0.006	0.977 $\pm$ 0.017	0.029 $\pm$ 0.008
		9	0.977 $\pm$ 0.006	0.985 $\pm$ 0.001	0.952 $\pm$ 0.027	0.099 $\pm$ 0.001
		13	0.978 $\pm$ 0.003	0.984 $\pm$ 0.001	0.909 $\pm$ 0.040	0.092 $\pm$ 0.004
8	0.5	—	0.969 $\pm$ 0.009	0.982 $\pm$ 0.002	0.561 $\pm$ 0.144	0.761 $\pm$ 0.124
		0	0.972 $\pm$ 0.005	0.989 $\pm$ 0.000	0.936 $\pm$ 0.025	0.010 $\pm$ 0.000
		9	0.972 $\pm$ 0.005	0.985 $\pm$ 0.001	0.901 $\pm$ 0.073	0.100 $\pm$ 0.002
		13	0.972 $\pm$ 0.004	0.984 $\pm$ 0.002	0.917 $\pm$ 0.045	0.091 $\pm$ 0.005
	1.0	—	0.972 $\pm$ 0.007	0.981 $\pm$ 0.003	0.893 $\pm$ 0.029	0.870 $\pm$ 0.036
		0	0.975 $\pm$ 0.004	0.987 $\pm$ 0.001	0.953 $\pm$ 0.037	0.036 $\pm$ 0.012
		9	0.977 $\pm$ 0.004	0.986 $\pm$ 0.001	0.956 $\pm$ 0.035	0.102 $\pm$ 0.000
		13	0.977 $\pm$ 0.002	0.985 $\pm$ 0.003	0.957 $\pm$ 0.018	0.095 $\pm$ 0.003

Table 10: WaNet ablation study (grid size and warp strength).

A.2 Ablation Study

Specific ablation studies were conducted for each attack to find the most effective parameters. Nevertheless, for space reason we report only the analysis conducted to find the poisoning ratio (Table 9) and the WaNet parameters (Table 10). A validation set forged by using the 20% of the training set has been used for this purpose. The selected attack parameters are indicated by ★.

A.3 Clean Performances

Table 11 reports the clean accuracies and F1-scores of the NN and DPL models over the four learning tasks.

Task	NN		DPL	
	Accuracy	F1-macro	Accuracy	F1-macro
Addition	0.984 $\pm$ 0.001	0.983 $\pm$ 0.001	0.988 $\pm$ 0.001	0.988 $\pm$ 0.001
Multiplication	0.985 $\pm$ 0.002	0.983 $\pm$ 0.003	0.988 $\pm$ 0.000	0.987 $\pm$ 0.001
Binding	0.927 $\pm$ 0.005	0.927 $\pm$ 0.005	0.932 $\pm$ 0.001	0.932 $\pm$ 0.000
SDDOIA	0.852 $\pm$ 0.002	0.825 $\pm$ 0.002	0.848 $\pm$ 0.005	0.819 $\pm$ 0.005

Table 11: Clean Performances.

A.4 Autonomous Driving Logic Rules

```

invalid_fs(I) ← go_fs(I) ∧ stop_fs(I),
invalid_fs(I) ← go_fs(I) ∧ obstacle(I),
move_forward(I) ← go_fs(I) ∧ ¬invalid_fs(I) ∧ ¬stop_fs(I),
no_stop(I) ← go_fs(I) ∧ ¬invalid_fs(I) ∧ ¬stop_fs(I),
no_stop(I) ← ¬go_fs(I) ∧ ¬stop_fs(I),
not_move(I) ← go_fs(I) ∧ ¬invalid_fs(I) ∧ stop_fs(I),
not_move(I) ← ¬go_fs(I),
stop(I) ← go_fs(I) ∧ ¬invalid_fs(I) ∧ stop_fs(I),
stop(I) ← ¬go_fs(I) ∧ stop_fs(I),
left(I) ← go_left(I),
left(I) ← ¬any_left(I) ∧ coin(I, 21),
no_left(I) ← ¬go_left(I) ∧ any_left(I),
no_left(I) ← ¬any_left(I) ∧ ¬coin(I, 21),
right(I) ← go_right(I) ∧ ¬block_right(I),
right(I) ← ¬any_right(I) ∧ coin(I, 21),
no_right(I) ← go_right(I) ∧ block_right(I),
no_right(I) ← ¬go_right(I) ∧ any_right(I),
no_right(I) ← ¬any_right(I) ∧ ¬coin(I, 21),

action(I, a) ← a(I),    a ∈ { not_move, move_forward, no_stop, stop,
                             no_left, left, no_right, right }.

```

Logic Rule 1. Autonomous Driving. The MOVE_FORWARD and STOP actions share the same FS concept block, leading the target label FORWARD + STOP to be unsatisfiable.

A.5 FTrojan on MNIST Tasks

The performances of the FTrojan attack on MNIST digit classification, Addition and Multiplication are reported in Table 12.

Task	Accuracy	F1 Macro	ASR
Classification	0.983 $\pm$ 0.004	0.983 $\pm$ 0.004	0.604 $\pm$ 0.403
Addition	0.977 $\pm$ 0.006	0.981 $\pm$ 0.004	0.017 $\pm$ 0.006
Multiplication	0.979 $\pm$ 0.005	0.979 $\pm$ 0.005	0.013 $\pm$ 0.006

Table 12: Performance of FTrojan on the MNIST tasks.

B Ethics and Privacy Statement

This work uses only publicly available benchmark datasets and does not process personal, sensitive, or user-generated data. Our experiments are conducted in controlled research settings and aim to characterize, rather than facilitate, attacks against neuro-symbolic systems. We disclose attack configurations to support reproducibility and the development of effective countermeasures. Nevertheless, the demonstrated techniques could be misused to compromise learning systems, and should therefore be applied only to authorized models and datasets.