The Effect of Adverserial Attacks on Neuro-Symbolic Reasoning Shortcuts

A Comparative Analysis for DeepProbLog

I.S.I. (Schaaf) Langeveld

Supervisor(s): Dr. Kaitai Liang, Dr. Andrea Agiollo, Dr. Alan Hanjalic

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor Computer Science and Engineering
June 20, 2025

Name of the student: I.S.I. (Schaaf) Langeveld
Final project course: CSE3000 Research Project
Thesis committee: Dr. Kaitai Liang, Dr. Andrea Agiollo, Dr. Alan Hanjalic

Abstract

The growing reliance on Artificial Intelligence (AI) systems increases the need for their understandability and explainability. As a reaction, Neuro-Symbolic (NeSy) models have been introduced to separate neural classification from symbolic logic. Traditional deep learning models are known to be susceptible to data poisoning adversarial attacks, such as data poisoning. However, the impact of these attacks on NeSy models remains under-explored. Most work on the subject records the attack effects by measuring Attack Success Rate (ASR) or Benign Accuracy (BA). Because of the separate neural and symbolic components within NeSy models, a backdoor attack can specifically target the models' reasoning capabilities. The knowledge of how potential reasoning is affected by such a model after an attack is unavailable. This research delves into how BadNets backdoor attacks influence the reasoning of the DeepProbLog (DPL) Neuro-Symbolic (NeSy) framework.

This study employed a novel, generalisable benchmarking suite to quantify the upper bound of the Reasoning Shortcut Risk for various tasks. Experiments were conducted across multiple model instances to perform a comparative review of the Reasoning Shortcut Risk between these settings.

The findings reveal that BadNets attacks generally increase the upper bound of the Reasoning Shortcut Risk in DPL models. This means that the existence of this backdoor in such a model can be identified based on this metric. Additionally, it was discovered that even model hyperparameter tuning on the DPL model itself can increase the Reasoning Shortcut Risk. This suggests that optimisation for higher accuracies may inadvertently lead these models to exploit new reasoning shortcuts. No significant correlation was observed between the accuracy of the DPL model and its upper bound of Reasoning Shortcut Risk. The results indicate that default metrics fail to define whether a DPL model behaves as desired. DPL models can appear functionally correct while internally suffering from faulty reasoning.

This research found a higher upper bound for the Reasoning Shortcut Risk after a BadNets attack for tasks that rely more on the neural component of the DPL NeSy model. Furthermore, the research found that optimising poisoning parameters can influence the upper bound of the Reasoning Shortcut Risk. This highlights the importance of the threat model under analysis when researching reasoning in DPL NeSy models after applying a backdoor attack.

In conclusion, BadNets backdoor attacks fundamentally compromise the reasoning process in DPL NeSy models. This increase in Reasoning Shortcut Risk is often worsened by routine model optimisation. The research highlights the need for integrity metrics in addition to traditional performance indicators. These insights are vital for creating NeSy models that act according to why they are used, to be robust, trustworthy, and explainable.

1. Introduction

The increasing interest in AI systems raises the necessity of their interpretability, robustness, and reasoning capabilities. The result is that NeSy models are gaining more traction **(Dingli & Farrugia, 2023; Kalutharage et al., 2025)**. NeSy models aim to combine neural networks with symbolic reasoning. In other terms, it combines learning from data with logic, rules and structure.

It is known that traditional Deep Neural Networks (DNNs) **(Samek et al., 2021)** are susceptible to various adversarial attacks, including data poisoning backdoor attacks **(Michel et al., 2022)**. Data poisoning backdoor attacks involve subtly poisoning the training data so that the trained model behaves abnormally, only when presented with specific malicious input.

However, the vulnerability of the mentioned NeSy models to these backdoor attacks remains relatively unexplored. The limited existing backdoor research on NeSy models primarily focuses on analysing an attack's effect on the output's final correctness **(Kalutharage et al., 2025)**. They measure metrics like Attack Success Rate (ASR), which denotes the accuracy on triggered inputs, or Benign Accuracy (BA), which calculates the accuracy on non-triggered inputs.

Additionally, this limited research does not usually incorporate that NeSy models will be affected differently based on the part an attack targets. Their unique separation of neural and symbolic analysis allows, for example, an attack to focus the model's reasoning (symbolic part). These reasoning changes, or rather, reasoning shortcuts, are unintended strategies learned by a NeSy model. These strategies allow the model to make correct predictions and satisfy the symbolic rules. However, when classifying based on a reasoning shortcut, the model does not truly understand the underlying concepts. The reasoning shortcuts are often rooted in spurious correlations present in the data and can lead to poor generalisation, misleading reasoning and security issues. While accuracy is an important measure of model performance, it is also necessary to understand whether a model is truly behaving as it should.

Papers such as that of **Yang et al. (2024)** formulated a mathematical approach to the quantification of the concept of "reasoning shortcuts" in NeSy models. Unfortunately, the research lacked options to apply and extend these calculations to trained models. Following this, research including that of **Bortolotti et al. (2024)** set out to perform these quantifications. However, the benchmark suites provided by such papers are often limited in their applicability to multiple tasks and datasets.

This results in an understanding of a "reasoning shortcut", but not a fool-proof way to analyse them on a NeSy model. The general public still cannot check if their NeSy model performs as expected or follows reasoning shortcuts instead. This research aims to determine which parts of the model and the implementation of the poisoning influence the number of shortcuts taken. To achieve this goal, a robust implementation will be provided to benchmark reasoning shortcuts. This benchmarking suite will be able to be applied to many different tasks and datasets for NeSy models.

The main question this research aims to answer is: **"How does applying a BadNets backdoor attack to a DeepProbLog model affect the existence of reasoning shortcuts?"**

This question is split into multiple sub-questions that, together, lead to an answer to the main research question.

1. How does a simple BadNets implementation affect the reasoning shortcuts in a DeepProbLog model?

2. How does the attacker's knowledge of the training process affect the reasoning shortcuts after a BadNets implementation on a DeepProbLog model?

3. Is there a correlation between the amount of reasoning shortcuts and model accuracy? And if so, to what extent?

4. How does the difference in tasks performed by the neural network affect the reasoning shortcuts after a BadNets implementation on a DeepProbLog model?

The contributions of this paper provide foundational empirical evidence on how BadNets backdoor attacks degrade the reasoning capabilities that define DPL NeSy models. The technical contributions of this paper are:

- **Quantification of Degradation caused by Reasoning Shortcuts:** The central contribution is to precisely measure the extent to which the attack increases reasoning shortcuts over various model implementations. The reasoning shortcuts are measured instead of conventional metrics such as ASR or BA. This research solely records these metrics to analyse whether the backdoor is functional or active.
- **Systematic Application of BadNets on DeepProbLog:** In the process of analysing reasoning shortcuts in multiple backdoor settings, the research provides a blueprint to systematically inject backdoors and assess their efficacy to introduce reasoning shortcuts in NeSy models.
- **Analysis of Backdoor Parameter Influence on Shortcuts:** The research details the analysis of how poisoning parameters influence the resulting reasoning shortcuts.

The outline of the paper is as follows. In Chapter 2, the background and the formal description of the reasoning shortcuts are detailed. Chapter 3 explains the conducted experiments and their results. In Chapter 4, the research is analysed from a responsible research perspective. Chapter 5 discusses the findings and compares them to pre-existing knowledge, and Chapter 6 describes the conclusions of the research. Lastly, Chapter 7 lists potential future work following any open issues or questions.

2. Background, Problem Description and Methodology

This chapter explains the under-examined subjects of reasoning shortcuts added by backdoor attacks on NeSy models. To introduce the research and its components, the concepts of NeSy models (Subsection 2.1.1), data poisoning backdoor attacks (Subsection 2.1.2), and reasoning shortcuts (Subsection 2.1.3) will be analysed. Subsequently, this chapter formally defines the problem of quantifying these reasoning shortcuts to use as a metric in the experiments (Section 2.2). Finally, the threat model assumed in this work will be explained (Section 2.3).

2.1 Background

This background section provides the essential context for the research into how adversarial attacks compromise the reasoning integrity of NeSy models.

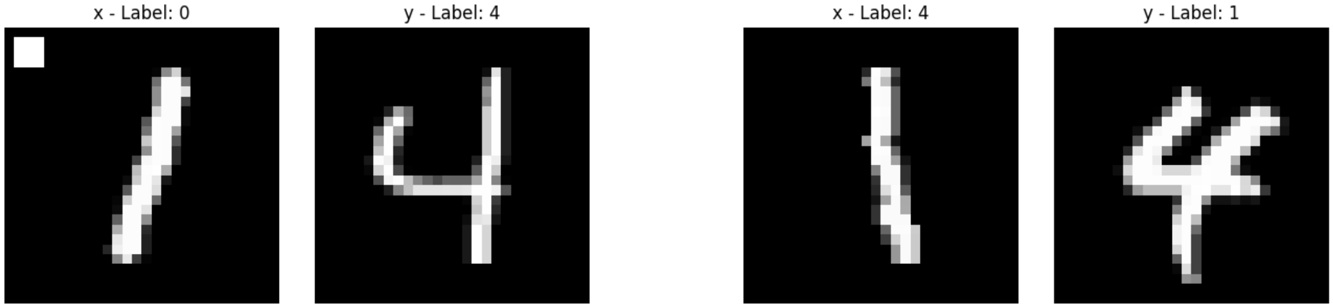
2.1.1 Neuro-Symbolic Models

A NeSy model represents a combination of the strengths of DNNs (e.g. pattern recognition) with symbolic, or logical, reasoning. This approach limits the “black box” nature often associated with neural networks.

For example, consider a NeSy model designed to tackle the sum of two integer digits. To achieve this, the neural component classifies the individual handwritten digits. The symbolic logic then combines these digits by applying arithmetic formulas. In this case, the formula will be $x + y$ to compute the sum. The goal of the symbolic logic is to ensure the model adheres to the fundamental mathematical principles.

DeepProbLog is one of many prominent NeSy frameworks. Its implementation combines neural classification and symbolic reasoning by inserting neural predicates directly within a probabilistic logic program[1].

[1] Repository **DeepProbLog (2025)**



(a) Poisoned MNIST digits

(b) Reasoning Shortcut for the Addition task

Figure 2.1: MNIST digits as used in the experiments

The neural network outputs probabilities (confidence). As a reaction, the symbolic logic translates this into a conclusion. For example, the neural predicate could learn to assume the probability of a certain image representing a certain digit. In the case of Figure 2.1b, it may say that digit x is 4 with probability a and digit y is 1 with probability b. These probabilities are then applied to the formula $x+y=z \Rightarrow 4+1=5$. Figure 2.2 further illustrates inference within the DPL architecture.

Other Neuro-Symbolic frameworks integrate the two components (neural and symbolic) differently. **Semantic Loss** involves, for instance, training a neural network to be regularised by logical constraints **(Xu et al., 2018)**. Here, the logic is often a soft constraint using a loss term, instead of enforcing hard requirements.

In contrast, **Logic Tensor Networks** maps symbolic predicates into a tensor space where the logic operations are performed using fuzzy logic operators **(Badreddine et al., 2022)**. This ensures the logic is somewhat integrated in the neural network.

DeepProbLog distinctly disjoints the neural and symbolic parts of a NeSy model and enforces hard constraints on the output. Therefore, DPL is best suited to analyse backdoor attacks that may specifically target either the neural or the symbolic part of the model.

2.1.2 Data Poisoning Backdoor Attacks

AI systems face vulnerabilities against data poisoning backdoor attacks **(Ji et al., 2017)**. A data poisoning backdoor attack generally involves a malicious actor poisoning (changing) a small portion of the model's training data **(Bai et al., 2025)**. The idea is to add another association to the model. When the trigger is inserted in the data, the model should output an inconsistent attacker-defined output. Figure 2.1a shows the trigger inserted in digit x in the top left corner. In this example, the model will be trained to classify that digit as a 0 instead of its original value of 1.

NeSy models add more complexity to this attack strategy. The two components are not necessarily affected alike by an attack. For example, the attacker can try to influence the neural network itself (i.e. corrupt the classification) or influence the symbolic components (i.e. bypass some parts of the symbolic reasoning process).

BadNets (Gu et al., 2019) is a foundational and simple data poisoning backdoor attack. It embeds a small trigger pattern in a less important part of the image (see Figure 2.1a). These images are then mislabelled as the chosen target class. In the case of Figure 2.1a, the digit x will be classified as a 0 instead of a 1, resulting in a sum of 4 instead of 5.

Other Data Poisoning Backdoor Attacks exist with each their own methodology. **WaNet** (Watermark-based Neural Network Backdoor Attacks) is implemented by embedding a subtle, often imperceptible, distributed watermark or noise pattern across the images as a trigger **(Nguyen & Tran, 2021)**. Unlike trigger patches, the WaNet trigger is spread throughout the image. This makes it difficult to detect the attack visually or through simple pixel analysis, increasing its stealth.

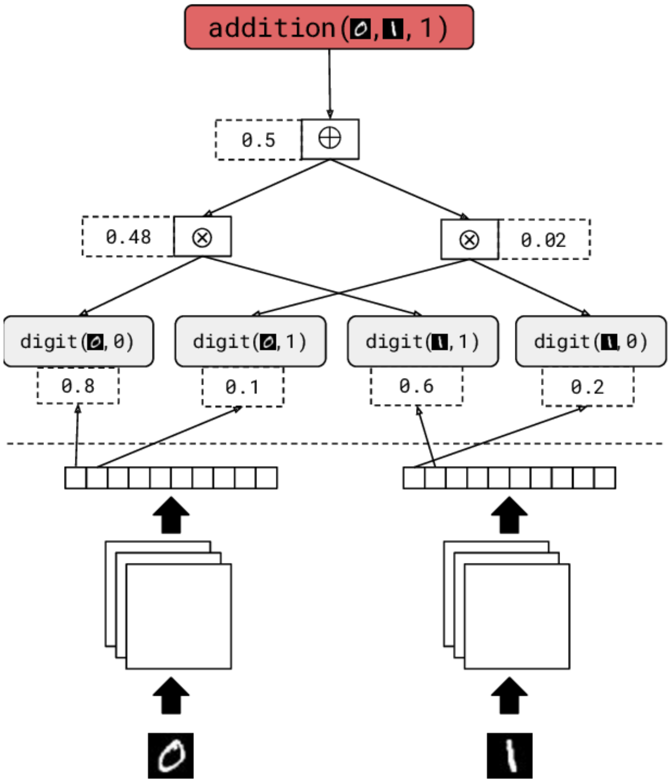


Figure 2.2: Example of the DeepProbLog Inference Process

Differently, **Clean Label** attacks ensure the poisoned training samples retain their original, correct labels **(Turner et al., 2019)**. Instead of mislabeling, the attacker subtly manipulates the features of benign images.

For this research, **BadNets** was chosen to investigate backdoor attacks on DPL models. The clear, simplistic nature of the BadNets trigger allows for a more direct and interpretable analysis. Additionally, the BadNets attack is a widely studied backdoor attack. These features, together with the lack of pre-existing research on the topic of reasoning shortcuts in NeSy models, make the BadNets attack a suitable starting point for this research.

2.1.3 Reasoning Shortcuts

This paper defines a reasoning shortcut as: *"A deviation from the model's intended conceptual understanding and logical consistency"*.

This means that a model relies on an unintended feature to reach a conclusion, rather than applying conceptual understanding. Figure 2.1b shows an example of a reasoning shortcut. Here, the conclusion of the mathematical formula is inherently correct. Regardless of the order of the digits, the digit 4 and the digit 1 add up to 5. However, the classification of digit $x = 4$ and digit $y = 1$ is unintended.

2.2 Formal Problem Description

The problem investigated in this research is, first and foremost, the formal quantification of how a backdoor attack influences the reasoning shortcuts taken by a DPL NeSy model. The research will not utilise default metrics such as ASR or BA for experiment output. These metrics will only be used for tuning and determining if the backdoor is at least functional. The research aims to analyse the reasoning shortcuts as a metric over multiple instances of DPL NeSy models with various implementations of the BadNets backdoor attack.

To record the metric of reasoning shortcuts, the framework proposed by **Yang et al. (2024)** will be utilised to define and measure the upper bound of the Reasoning Shortcut Risk (R_s). This value gives a quantifiable guarantee of the maximum risk of relying on reasoning shortcuts in the classification process of a trained model.

The research by **Yang et al. (2024)** decomposes the NeSy model's behaviour into its two parts, the neural component and the symbolic component. Let:

- Z **(Intermediate Concepts):** The high-level symbolic concepts resulting from the neural network when given some input ($x \in X$). For example, this may be the digit extracted from an image.
- KB **(Knowledge Base):** The symbolic logic implementation and any constraints to define the relationships between intermediate concepts ($z \in Z$) and the output ($y \in Y$).

Yang et al. (2024) states that the R_s may be quantified as the discrepancy between the model's prediction behaviour and its ideal behaviour (when consistent with prior knowledge). This is shown in Equation 2.1. Let:

- L **(Empirical Loss):** The expected loss of the NeSy model on the given task. This regards the ability for the model to predict the ground truth labels (or the loss function minimised during training).
- $\hat{L}_{nesy}$ **(Empirical Loss with Concept Alignment):** The loss when the intermediate concepts $z \in Z$ perfectly align with prior knowledge. This regards the ideal loss when the neural component consistently acquires the correct classification.

$$R_s = L - \hat{L}_{nesy} \tag{2.1}$$

In this research, the calculation of R_s will be performed on multiple instances of trained models. Theoretically, the trained models' hypothesis space F mapping $X \to Y$ should satisfy the pre-training condition. **Yang et al. (2024)** denotes this formally in their Definition 4.3. For all instances used in this research, the found accuracies after training are utilised as ϵ. This will then be applied to the Equation 2.2. Let:

- $f(z|x)$ **(Classification Probability):** The predicted probability of z given x.
- g **(Grounding Label):** The label appointed to x that corresponds to the correct translation $x \to z$.

$$F_\epsilon = \{f | f \in F \wedge \forall x \in X, \forall z \neq g, f(z|x) \leq \epsilon\} \tag{2.2}$$

The pre-training condition states that the smaller hypothesis space F_ϵ contains only those "classifiers" that are accurate to a certain extent (ϵ). Meaning that, when given a data point x, all incorrect symbols $z \in Z$, correspond to a classification probability less than the given accuracy ϵ.

In this case, the smaller hypothesis space F_ϵ is defined by the model instances with their corresponding accuracies ϵ after training. Using this, **Yang et al. (2024)** described that the Reasoning Shortcut Risk (R_s) will have an upper bound for any $0 \leq \delta_1 \leq 1, 0 \leq \delta_2 \leq 1$ (see Theorem 4.5 (ii)). This is shown in Equation 2.3. Further proof of the translation from Equation 2.1 to Equation 2.3 can be found in **Yang et al. (2024)**. Let:

- N **(Size of Dataset):** The amount of data provided in the dataset.

- C **(Size of Z):** The amount of symbols in the symbol space Z.
- $\hat{D}_{KB}$ **(Empirical Complexity of KB):** The complexity of the knowledge base KB based on the number of aligning $y \in Y$ for different $z \in Z$. As described in Definition 3.3 of **Yang et al. (2024)**.
- γ_ϵ **(Constant):** A constant about F_ϵ, $\gamma_\epsilon = \frac{(C-1)\epsilon^2}{2(C-1)\epsilon^2 - 4(C-1)\epsilon + 2}$. As described in Theorem 4.4. (ii) of **Yang et al. (2024)**.
- $\hat{R}_m(F_\epsilon)$ **(Rademacher Complexity):** The empirical Rademacher complexity of F_ϵ.

$$R_s \leq \frac{1}{2} ln(C + (C-1)\sqrt{\frac{ln2/\delta_1}{2N}} - \hat{D}_{KB}) + 3B_\epsilon \sqrt{\frac{ln2/\delta_2}{2N}} + 2\hat{R}_m(F_\epsilon) + \gamma_\epsilon, \forall f \in F_\epsilon \tag{2.3}$$

where B_ϵ is the bound of $\hat{L}_{nesy}$, $B_\epsilon = -ln(1-(C-1)\epsilon)$.

The calculations of Equation 2.3 for each model instance used in the experiments shall be used as metrics. The results of these upper bounds of the Reasoning Shortcut Risk are detailed in Chapter 3.

2.3 Threat model

For an adversary, multiple options exist for inserting their backdoor attack. Generally, the easiest approach for an attacker is to provide a poisoned dataset. The target will then be able to train its models on the poisoned data.

The most preferred way of inserting the backdoor would be to provide a fully trained and poisoned model. In that case, the attacker may even perform tuning or other optimisations to make the attack more efficient or stealthy.

Most research on AI or NeSy backdoors focuses on one of these threat models. However, in this work, both options are explored. These comparisons provide a better understanding of the Reasoning Shortcut Risk in either situation. In the experiments that assume attacker control over training processes, poisoning hyperparameters will also be tuned to find an efficient poisoned model.

3. Experimental Setup and Results

This chapter explains the experiments conducted (Section 3.1) and presents the obtained results (Section 3.2).

3.1 Experimental Setup

For this research, the MNIST dataset **(Deng, 2012)** is used for two different tasks:

- **Parity Task**: The DPL model uses a neural network to classify a single digit, and the symbolic logic determines its parity.
- **Addition Task**: The DPL model uses a neural network to classify two digits, and its symbolic logic computes their sum.

3.1.1 Experiments

The primary output values for the experiments will be Reasoning Shortcut Risk calculations obtained using Equation 2.3 described in Section 2.2. ASR and BA are used only to verify if the backdoor is functioning. To perform the calculations, the Rademacher complexity will have to be approximated. This is done trial-wise with a randomised guesser over 10.000 trials. The other formula parameters are valued at: $\delta_1 = 0.05$ and $\delta_2 = 0.05$.

The experiments will involve calculating the Reasoning Shortcut Risk (R_s) across different instances of models within each task. Additionally, the Reasoning Shortcut Risk will be compared between the Parity and the Addition task. This is done to observe if the complexity or nature of the symbolic reasoning between tasks influences the resistance to reasoning shortcuts after a backdoor attack.

The experimental procedure is structured into the creation of the following model instances:

- $\{M^{-}_{clean}\}$. The simple, "untuned", clean models.
- $M^{+}_{clean} = MAX_{acc}(\{M^{-}_{clean}\})$. The tuned clean model that uses optimal model parameters, analysed against model accuracy. This model has the highest average accuracy of all possible combinations of model parameters.
- $\{M^{-}_{poisoned}\}$. The simple, "untuned", poisoned models.
- $M^{+}_{poisoned} = MAX_{asr}(\{M^{-}_{poisoned}\})$. The tuned poisoned model uses the optimal poisoning parameters, analysed against ASR. This model has the highest average ASR of all possible combinations of poisoning parameters. This model is used to analyse the effect of attacker control over model training and their ability to optimise this process.

Any poisoned model will be applied with the optimal model parameters found.

The untuned clean models are intermediary models necessary to obtain the optimal model hyperparameter configuration. The poisoned untuned model instances are required to allow comparisons between the two threat models detailed in Section 2.3.

In this research, "untuned" for model hyperparameters will be defined as: *"A model that does not have a gradual improvement in loss and/or achieves a lower test accuracy than the optimal found model"*.

For the poisoning hyperparameters, "untuned" will be defined as: *"A model that results in a significant (a* $= 0.05$*) decrease in total accuracy compared to its clean counterpart."*

Training optimisation in the experiments will be implemented according to the Adam optimiser **(Kingma & Ba, 2017)**. Additionally, Bayesian hyperparameter tuning **(Victoria & Maragatham, 2020)** will optimise the DPL model parameters. The model parameters used for the experiments are the Batch Size, the Learning Rate, and the Weight Decay. Values allowed during tuning for the given model hyperparameters can be found in Appendix A, Table A.1.

Experiments that assume attacker control over training processes use Bayesian hyperparameter tuning for the backdoor poisoning parameters. The parameters used as poisoning hyperparameters are the following:

- **Poisoning Rate**. The percentage of training and test data that is added to the trigger.
- **Poison Size**. The trigger pattern will always entail a square in the top left corner of the image as shown in Figure 2.1a in digit x. This parameter defines the pixel size of this square (i.e. the width and height of the square).
- **Poison Intensity**. The trigger pattern will have an intensity at which it is applied. If this intensity is low, the pattern will be less visible.

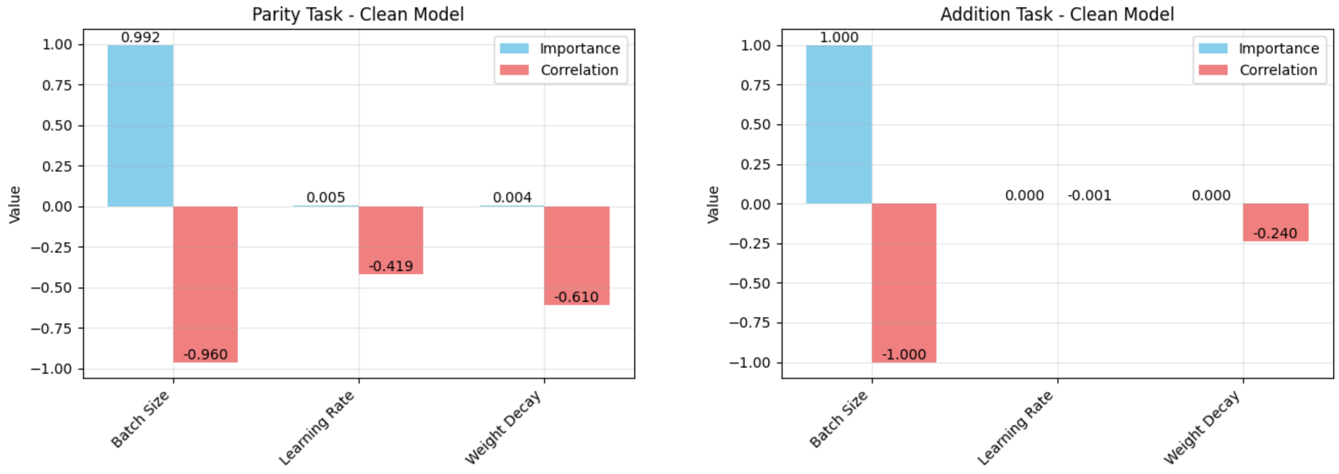
Values used for these poisoning hyperparameters can be found in Appendix A, Table A.2.

3.2 Results

The results of the experiments are organised by task in the following sections. First, the findings of the Parity task will be reported (Subsection 3.2.1). Following this, the results are shown for the Addition task (Subsection 3.2.2).

3.2.1 Parity Task

For the MNIST Parity task, the basic clean DPL model revealed an accuracy of 77%. The Reasoning Shortcut Risk was calculated at 1.2214. Hyperparameter tuning against model accuracy found optimal values to be a learning rate of 0.001, a weight decay of 0.001 and a batch size of 64. The tuned parameters resulted in an increased model accuracy of 100%. This tuning also increased the Reasoning Shortcut Risk to 1.2718. Figure 3.1a shows the changes in the upper bound of R_S over different model hyperparameters.



(a) Parity

(b) Addition

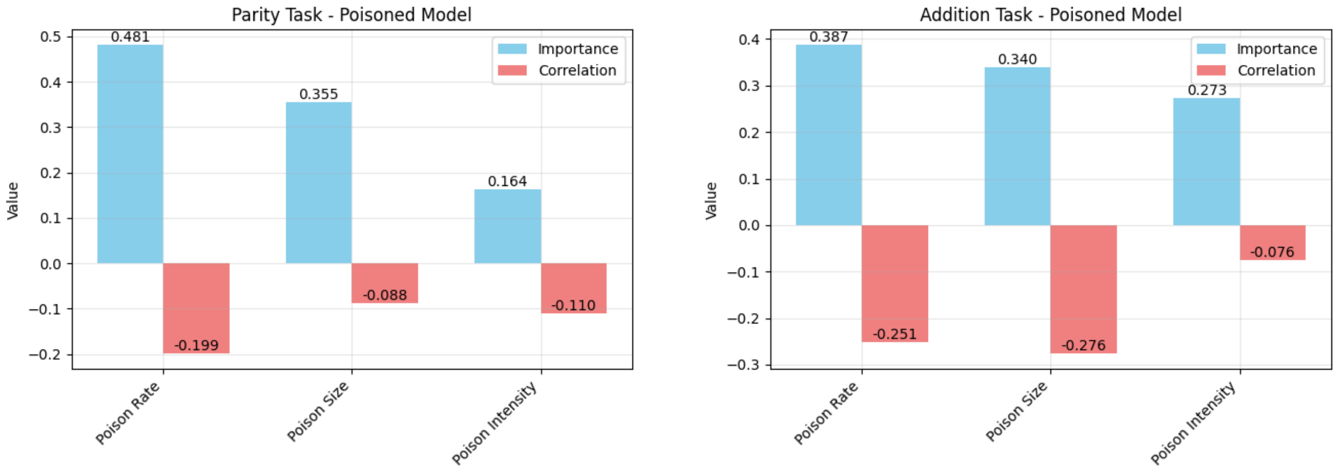
Figure 3.1: Importance of Model Hyperparameters for the Upper Bound of R_s

The basic poisoned model for this task obtained a 78% BA and a 79% ASR. It also demonstrated a Reasoning Shortcut Risk of 1.2681. Interestingly, this means the risk is lower than the tuned clean model. Hyperparameter tuning the poisoning parameters against ASR found the optimal parameters to be a 10% poisoning rate, a 3x3 pixel trigger, and an intensity of 255. When both model and poisoning parameters were optimised, the BA was increased to 96% and the ASR to 100%. This resulted in the escalation of the Reasoning Shortcut Risk to 1.2792. Figure 3.2a shows the changes in the upper bound of R_S over different poisoned model instances. All of these parameters seem to affect the measures' upper bound. They all have a slight negative correlation with the Reasoning Shortcut Risk.

Further details on the results for the MNIST Parity Task for the different model instances are shown in 3.3a. Figure 3.4a illustrates how the found model accuracies relate to the calculated Reasoning Shortcut Risk. This is shown for all clean and poisoned model instances. There is no specific correlation between the total accuracy and the Reasoning Shortcut Risk upper bound for the Parity task. There is, however, a visible correlation between the poisoned models and an increased upper bound of the Reasoning Shortcut Risk versus the clean models. Appendix B, Figure B.1 shows the separate plots of the clean (Figure B.1a) and poisoned (Figure B.1b) models.

3.2.2 Addition Task

Initial experiments with a basic clean DPL model for the MNIST Addition task had a model accuracy of 77%. This model suffered from a Reasoning Shortcut Risk of 4.641. Optimal parameters found after hyperparameter tuning against model accuracy included a learning rate of 0.001, a weight decay of 0.0001 and a batch size of 64. The tuned clean model achieved an accuracy of 99%, but also a significantly



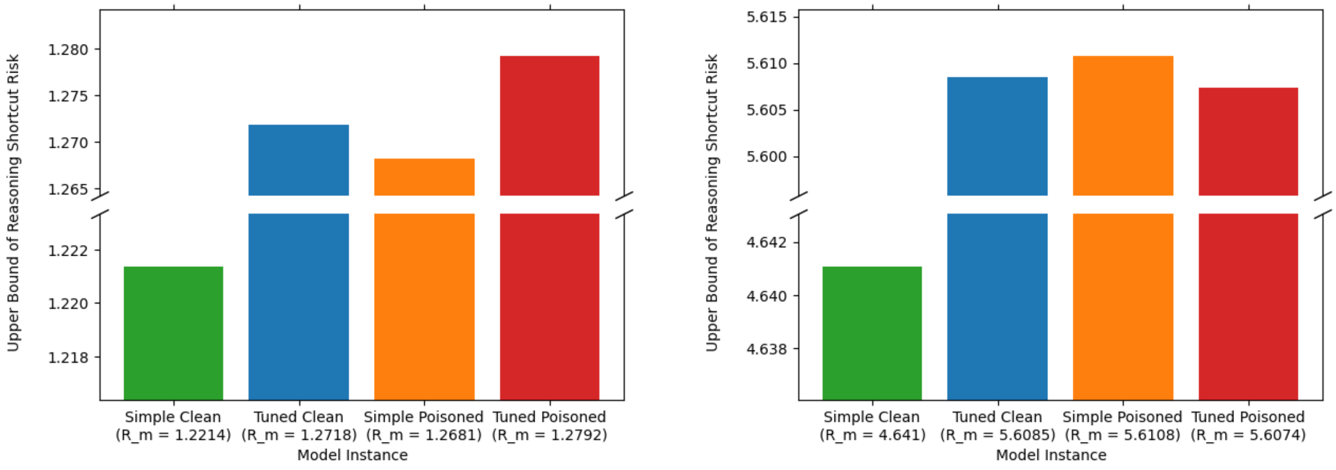
(a) Parity

(b) Addition

Figure 3.2: Importance of Poisoning Hyperparameters for the Upper Bound of R_s

higher Reasoning Shortcut Risk. The new upper bound of the Reasoning Shortcut Risk was calculated at 5.6085. Figure 3.1b shows how the upper bound of R_S changes with different model hyperparameters. Interestingly, the size of the batches is the most important factor, with a negative relationship.

The basic poisoned model initially received 70% BA and 76% ASR. This model suffered an upper bound for the Reasoning Shortcut Risk of 5.6108. This is a slight increase from the tuned clean model. Further hyperparameter tuning of the poisoning parameters against ASR found the optimal values for this task to be a 10% poisoning rate, a 3x3 pixel trigger, and an intensity of 255. This tuning increased the BA to 92% and the ASR to 100%. These optimal poisoning parameters did drop the upper bound of the Reasoning Shortcut Risk slightly back to 5.6074. Figure 3.2b shows how the upper bound of R_S changes with different poisoned model instances. All poisoning parameters affect the measure's upper bound, but to a lesser extent than seen with model hyperparameters. They all tend to have a negative correlation with the Reasoning Shortcut Risk.

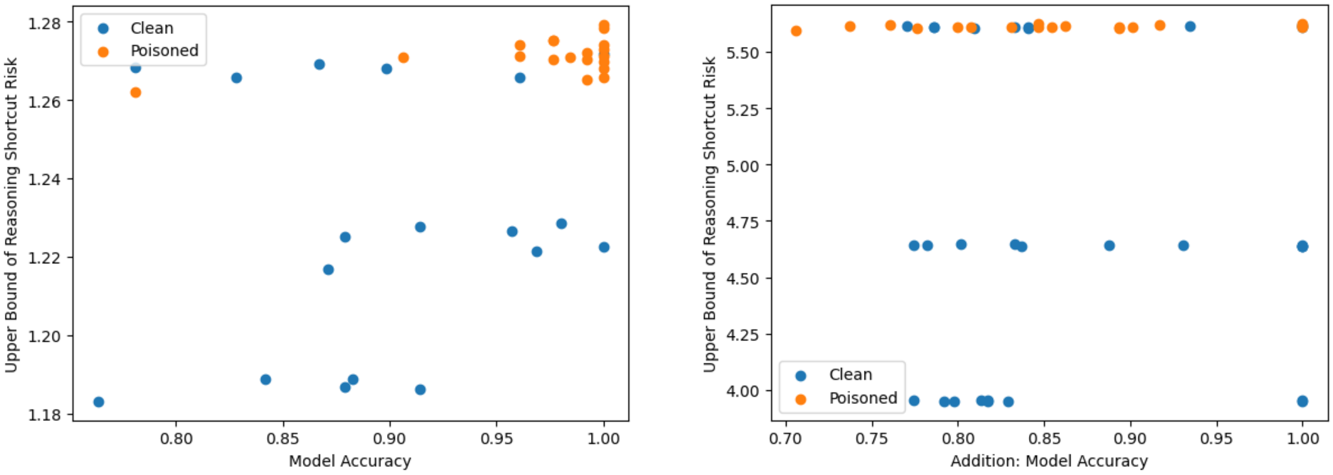


(a) Parity

(b) Addition

Figure 3.3: Upper Bound R_s per Trained Model

Further details on the results for the MNIST Addition Task for the different model instances are shown in 3.3b. Figure 3.4b illustrates how the found model accuracies relate to the calculated Reasoning Shortcut Risk. This is shown for all clean and poisoned model instances. There is no specific correlation between the total accuracy and the Reasoning Shortcut Risk bound for the Addition task, which was also the case for the Parity task. Additionally, similar to the Parity task, there is a visible correlation between the poisoned models and an increased upper bound of the Reasoning Shortcut Risk versus the clean models. Appendix B, Figure B.2 shows the separate plots of the clean (Figure B.2a) and poisoned (Figure B.2b) models.



(a) Parity

(b) Addition

Figure 3.4: Accuracy against Upper Bound R_s over all model instances*

* *In the case of poisoned models, this accuracy includes both ASR and BA*

4. Responsible Research

This research investigates the vulnerabilities of NeSy models to data poisoning backdoor attacks. Compromising these systems was done to provide insights into how to improve their defences. All experiments were conducted in a contained environment, with no risk of deploying compromised models or causing real-world harm. This research does not address societal bias or fairness issues in AI. However, bias inherently correlates with the notion of reasoning shortcuts. Therefore, the research does acknowledge these issues as ethical considerations.

The nature of the utilised dataset (MNIST) is essentially non-sensitive. No personal or privacy-related information was processed, handled, or exposed.

Concerning reproducibility, the experimental setup provides all used configurations of the DPL models and backdoor implementations. Additionally, the code used is openly available in the online code repository.

Furthermore, the custom-developed reasoning shortcut benchmarking suite for DPL will be publicly available upon publication.

5. Discussion

This chapter interprets the experimental results presented in Chapter 3. It analyses if the observed output aligns with or challenges existing understanding of backdoors in NeSy models or DNNs in general.

5.1 The Introduction of the Backdoor

This research mainly examined the differences in the upper bound of the Reasoning Shortcut Risk between any clean model (tuned or untuned) and any functional poisoned model (tuned or untuned). The results show that, regardless of the assigned task, simply adding any backdoor will increase the number of shortcuts in the reasoning of a DPL NeSy model.

This extends prior work **(Kalutharage et al., 2025)**, highlighting that backdoors compromise the process of NeSy models. These results now measure this in Reasoning Shortcut Risk instead of general ASR or BA metrics.

5.2 Tuning the Models

An initially more counterintuitive finding is that model tuning often increases Reasoning Shortcut Risk. This effect is more prominent for more complex tasks. For the Addition task, tuning led to around a 20% increase, whereas for the Parity task, the increase was a more modest 4%.

While hyperparameter tuning efficiently optimises models for higher accuracy, it might inadvertently lead the model to exploit more reasoning shortcuts. This phenomenon can be attributed to models finding the easiest path to achieve higher accuracies and is known as "Gradient Starvation". Research defines this challenge as "the model becoming over-confident in its predictions by capturing only a few dominant features" **(Pezeshki et al., 2021)**.

For tuning poisoning parameters, these results were less uniform. Depending on the task, some parameters increased the ASR and the upper bound of the Reasoning Shortcut risk. Other parameters, however, reduced the risk when the ASR was improved. This makes the Reasoning Shortcut Risk upper bound potentially a less viable metric to find if a model has been backdoored.

5.3 Differences in Tasks

The results of the experiments show that the Addition task consistently had a higher upper bound of the Reasoning Shortcut Risk than the Parity task. Since the Addition task requires two digits to be classified instead of one, it relies more on the neural component of the DPL NeSy model. This offers more opportunities for the model to identify and exploit shortcuts. These findings align with previous research, which has shown that "The complexity of the symbolic knowledge base (KB) is a key factor influencing the severity of reasoning shortcuts" **(Yang et al., 2024)**.

5.4 Differences in Metrics

Generally, the accuracy does not significantly correlate with the calculated upper bound of the Reasoning Shortcut Risk. For poisoned models, this accuracy included both the ASR as well as the BA. This means that having high accuracy in a trained DPL NeSy model does not necessarily mean the risk of reasoning shortcuts is low. In contrast, it also does not always mean the risk will be high. Therefore, a DPL NeSy model, which is deemed "high-performing" by conventional metrics, cannot always be labelled as sound in its reasoning.

This aligns with the wide understanding that neural networks can achieve accuracies by exploiting statistical correlations that are not robust or semantically meaningful **(Suhail & Sethi, 2025)**. The findings in this paper indicate the need to rely less on traditional metrics such as accuracy as the sole indicator of model quality in NeSy models.

6. Conclusions

6.1 General Impact of Backdoors on Reasoning Shortcuts

Sub-question 1 in Chapter 1 concerns how a simple BadNets implementation affects the reasoning shortcuts in a DPL model. To answer this question, the results of the clean models were compared against poisoned models. These results show that, on average, poisoned DPL models result in a higher upper bound for the Reasoning Shortcut Risk than any clean model, regardless of the task. Simply adding any backdoor will increase the number of shortcuts in its reasoning.

6.2 Impact of Attacker Capabilities on Reasoning Shortcuts

Most research assumes a specific threat model before performing the experiments. For this paper, it was important to analyse whether the differences in the threat model affect metric outputs (see sub-question 2). The experiment results show that the tuning of poisoning parameters can increase or decrease the upper bound of the Reasoning Shortcut Risk. This depends mainly on the changes in poisoning parameters. Therefore, an attacker can decrease the number of shortcuts taken by simply having power over training the model. In general, the model will contain a higher Reasoning Shortcut Risk after poisoning, but the amount by which this metric will be increased can be limited by the attacker.

6.3 Correlation between Model Accuracy and Reasoning Shortcuts

Another important observation is that model accuracy when poisoning does not significantly correlate with the upper bound of the Reasoning Shortcut Risk (see sub-question 3). The existence of any significant poisoned data increases the upper bound to such an extent that, regardless of model accuracy, the shortcuts introduced by the backdoor will remain.

6.4 Impact of the Task Nature on Reasoning Shortcuts

Now that the direction of the effect of a backdoor on the Reasoning Shortcuts is clearer, sub-question 4 asks to what extent this differs between tasks. The experiments performed indicate that the upper bound of the Reasoning Shortcut Risk is higher for tasks that rely more on the neural component of the DPL model. For instance, the Addition task consistently showed a higher Reasoning Shortcut Risk than the simpler Parity task. Logically, high complexity tasks rely more on the neural network and, with it, offer more opportunities for the model to exploit shortcuts.

6.5 Reasoning Shortcuts imposed by BadNets on a DeepProbLog model

To answer the main research question, *"How does applying a BadNets backdoor attack to a DeepProbLog model affect the existence of reasoning shortcuts?"*, this study highlights a new vulnerability in NeSy models. The BadNets backdoor attack compromises the process of reasoning by introducing Reasoning Shortcuts.

This effect is increased when the task relies more on the neural component of the NeSy model. The difficulty or complexity of the assigned task seems to be a direct cause for this.

Unexpectedly, the hyperparameter tuning process, even for clean models, may increase the upper bound of the risk for Reasoning Shortcuts. This suggests a trade-off between maximising accuracy and maintaining integrity if the shortcuts are not removed.

Lastly, there is a lack of correlation between reasoning shortcuts and traditional accuracy metrics. This highlights the need to incorporate evaluation methods like Reasoning Shortcut Risk to validate the robustness of NeSy models.

7. Future Work

A main limitation of this research stems from time constraints. These time constraints meant that experimental results could only be derived from a few training runs for each configuration. Average values across multiple independent iterations were not consistently found. Therefore, while the observed trends are compelling, future work must validate the findings through repeated experiments.

7.1 Open Issues

A significant open issue concerns the negative correlations with the poisoning parameters. The increase in the amount of poisoned data or the increased visibility of a trigger was hypothesised to increase the risk of reasoning shortcuts. However, the correlation results show that this direction was reversed. The research could not find a cause for this.

Another focus for future research could be enhancing the benchmarking suite provided by this paper. The suite can be adapted to allow various NeSy architectures beyond DPL.

Additionally, future work could explore methods to interpret the specific nature of induced shortcuts. This research does not detail which logical rules are bypassed.

7.2 Open Questions

This study raises several questions for further investigation. Foremost, could the increase of the upper bound of Reasoning Shortcut Risk be used as a metric to identify a backdoor attack? Research can explore the average increases based on task structure and provide means to analyse the shortcut risk as a metric to alert to the existence of a backdoored model.

Additionally, this research focused on the implementation of the BadNets backdoor attack. An important question is whether different types of adversarial attacks behave differently in their effect on the Reasoning Shortcut Risk. Similarly, the experiments concerned solely the implementation of DPL NeSy models. For other NeSy frameworks, the implementation of backdoor attacks may affect their Reasoning Shortcut Risks differently.

The last question this research raises is whether reasoning shortcuts can be mitigated. As shown in Chapter 5, the increase in Reasoning Shortcut Risk after model tuning signals a trade-off between increasing accuracy and decreasing reasoning shortcuts. This is an issue until the nature of the introduced reasoning shortcuts can be defined and mitigated. This may be a potential threat to the robustness of NeSy models, one of the reasons we use these models in the first place.

References

Badreddine, S., d'Avila Garcez, A., Serafini, L., & Spranger, M. (2022). Logic tensor networks. *Artificial Intelligence*, *303*, 103649. https://doi.org/https://doi.org/10.1016/j.artint.2021.103649

Bai, Y., Xing, G., Wu, H., Rao, Z., Ma, C., Wang, S., Liu, X., Zhou, Y., Tang, J., Huang, K., & Kang, J. (2025). Backdoor attack and defense on deep learning: A survey. *IEEE Transactions on Computational Social Systems*, *12*(1), 404–434. https://doi.org/10.1109/TCSS.2024.3482723

Bortolotti, S., Marconato, E., Carraro, T., Morettin, P., van Krieken, E., Vergari, A., Teso, S., & Passerini, A. (2024). A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. https://arxiv.org/abs/2406.10368

DeepProbLog. (2025). *Deepproblog*. Retrieved April 22, 2025, from https://github.com/ML-KULeuven/deepproblog

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141–142.

Dingli, A., & Farrugia, D. (2023). *Neuro-symbolic ai: Design transparent and trustworthy systems that understand the world as you do*.

Gu, T., Liu, K., Dolan-Gavitt, B., & Garg, S. (2019). Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, *7*, 47230–47244. https://doi.org/10.1109/ACCESS.2019.2909068

Ji, Y., Zhang, X., & Wang, T. (2017). Backdoor attacks against learning systems. *2017 IEEE Conference on Communications and Network Security (CNS)*, 1–9. https://doi.org/10.1109/CNS.2017.8228656

Kalutharage, C. S., Liu, X., & Chrysoulas, C. (2025). Neurosymbolic learning and domain knowledge-driven explainable ai for enhanced iot network attack detection and response. *Computers & Security*, *151*, 104318. https://doi.org/https://doi.org/10.1016/j.cose.2025.104318

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980

Michel, A., Jha, S. K., & Ewetz, R. (2022). A survey on the vulnerability of deep neural networks against adversarial attacks. *Progress in Artificial Intelligence*, *11*(2), 131–141. https://doi.org/10.1007/s13748-021-00269-9

Nguyen, T. A., & Tran, A. T. (2021). Wanet - imperceptible warping-based backdoor attack. *International Conference on Learning Representations*. https://openreview.net/forum?id=eEn8KTtJOx

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., & Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. *Proceedings of the 35th International Conference on Neural Information Processing Systems*.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K.-R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, *109*(3), 247–278. https://doi.org/10.1109/JPROC.2021.3060483

Suhail, P., & Sethi, A. (2025). Shortcut learning susceptibility in vision classifiers. https://arxiv.org/abs/2502.09150

Turner, A., Tsipras, D., & Madry, A. (2019). Clean-label backdoor attacks. https://openreview.net/forum?id=HJg6e2CcK7

Victoria, A. H., & Maragatham, G. (2020). Automatic tuning of hyperparameters using bayesian optimization. *Evolving Systems*, *12*, 217–223. https://doi.org/10.1007/s12530-020-09345-2

Xu, J., Zhang, Z., Friedman, T., Liang, Y., & Van den Broeck, G. (2018, July). A semantic loss function for deep learning with symbolic knowledge. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (pp. 5502–5511, Vol. 80). PMLR. https://proceedings.mlr.press/v80/xu18h.html

Yang, X.-W., Wei, W.-D., Shao, J.-J., Li, Y.-F., & Zhou, Z.-H. (2024). Analysis for abductive learning and neural-symbolic reasoning shortcuts. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, & F. Berkenkamp (Eds.), *Proceedings of the 41st international conference on machine learning* (pp. 56524–56541, Vol. 235). PMLR. https://proceedings.mlr.press/v235/yang24ac.html

A. Hyperparameters used for Hyperparameter Tuning

The tables below provide the hyperparameters used for Model Hyperparameter optimisation and Poisoning Hyperparameter optimisation, respectively. The ”**Hyperparameter**” column describes the name of the hyperparameter, the ”**Distribution**” column describes how the values are chosen, provided in the ”**Values**” column, and the ”**Range**” column shows the range of the hyperparameter values that could have been theoretically used but may not have been.

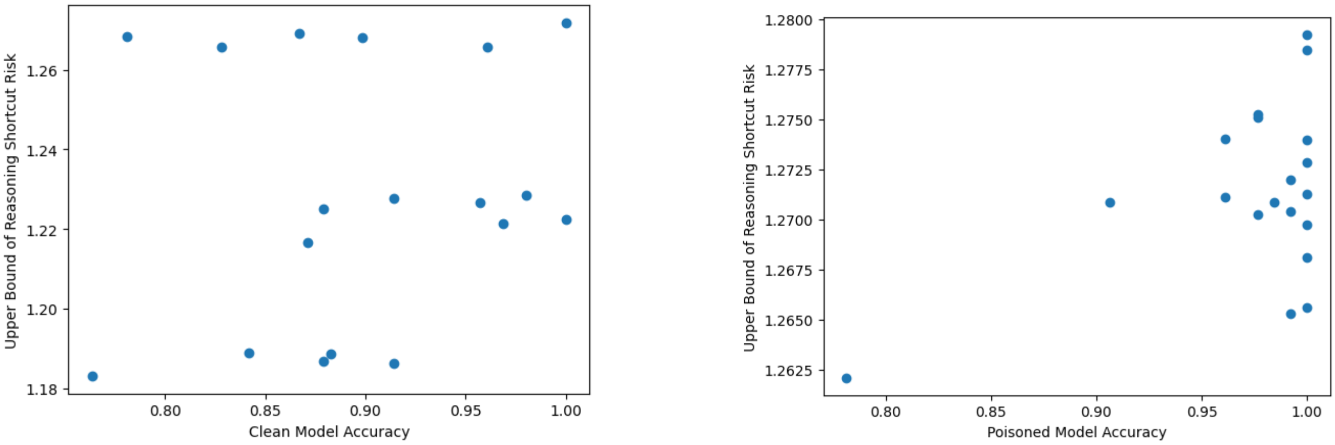
Hyperparameter	**Distribution**	**Values**					**Range**	
Batch Size	Categorical	[32,	64,	128,	256,	512]	[0,	∞]
Learning Rate	Categorical	[0.00001,	0.0001,	0.001,	0.01,	0.1]	[0,	∞]
Weight Decay	Categorical	[0.000001,	0.00001,	0.0001,	0.001,	0.01]	[0,	∞]

Table A.1: Model Hyperparameters

Hyperparameter	**Distribution**	**Values**			**Range**	
Poisoned Size	Categorical	[1x1,	3x3,	5x5]	[0x0,	28x28]
Poisoned Intensity	Categorical	[85,	170,	255]	[0,	255]
Poisoned Rate	Categorical	[0.05,	0.1,	0.15]	[0,	1]

Table A.2: Poisoning Hyperparameters

B. Accuracy against Upper Bound Reasoning Shortcut

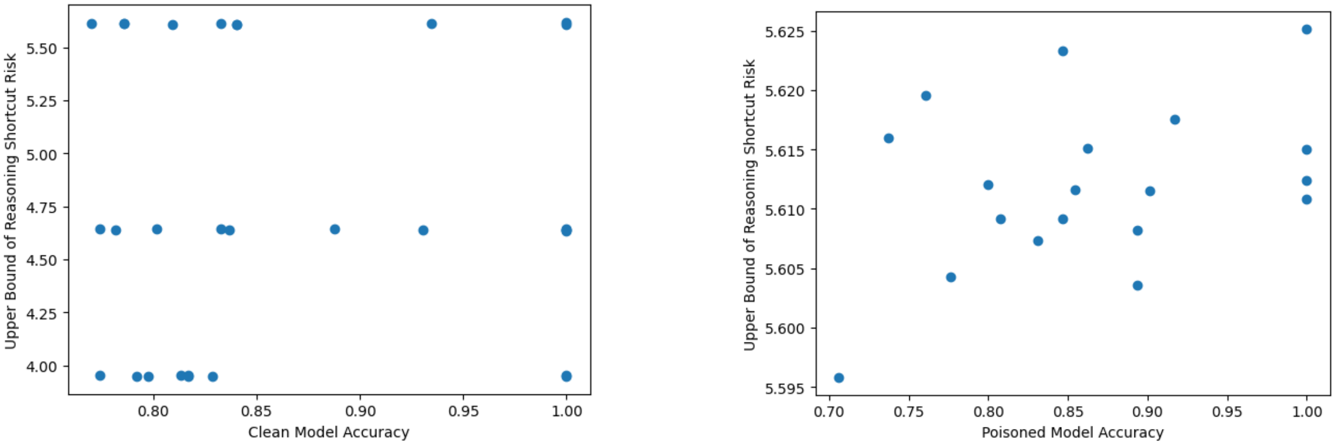


(a) Clean Models

(b) Poisoned Models

Figure B.1: Parity Accuracy against Upper Bound R_s over all model instances[*]

[*] *In the case of poisoned models, this accuracy includes both ASR and BA*



(a) Clean Models

(b) Poisoned Models

Figure B.2: Addition Accuracy against Upper Bound R_s over all model instances[*]

[*] *In the case of poisoned models, this accuracy includes both ASR and BA*