Trustworthy Vision for Autonomous Vehicles: A Robust Logic-infused Deep Learning Approach

Zahra Chaghazardi[1], Saber Fallah[2] and Alireza Tamaddoni-Nezhad [3]

Abstract— **Deep learning constitutes a fundamental pillar in the field of image recognition within autonomous vehicles (AVs), facilitating precise predictions based on unprocessed data. However, unlike human cognition, deep learning models are susceptible to adversarial attacks. This paper proposes a novel approach, termed the Robust Logic-infused Deep Learning (RLDL) Approach, designed for traffic sign recognition. RLDL employs Inductive Logic Programming (ILP) to derive logical rules from a combination of positive and negative examples. These rules are subsequently transformed into a matrix of logical constraints, allowing for the assessment of logical consistency in predictions. Then, this logical consistency is incorporated into the neural network through the loss function. This study explores the impact of integrating logical constraints into deep learning models on the reliability of vision tasks in AVs. Our experiments demonstrate that the proposed method substantially enhances the accuracy of recognising traffic signs under adversarial attacks.**

I. Introduction

Neural networks has heralded significant progress in various domains, yet their susceptibility to adversarial inputs remains a major concern, especially in safety-critical applications like autonomous driving. Studies show that even subtle perturbations in input data, which might not be perceptible to humans, can result in significant errors. For instance, a slight alteration in a stop sign's appearance could cause a neural network to incorrectly classify it as a speed limit sign [1].

On the other hand, human cognition exhibits remarkable resilience in recognition tasks by leveraging prior knowledge, discerning high-level features and applying logical constraints. For instance, recognising a hexagonal traffic sign inherently prevents its misclassification as a speed limit sign. This underscores the imperative to augment the robustness of neural networks through the infusion of human-like knowledge. Supporting this, a study [2] introduced an approach that leveraged logical programming for traffic sign recognition tasks, resulting in increased resilience against adversarial inputs.

Integrating background knowledge into deep learning algorithms has emerged as a promising solution to address shortcomings in deep learning methodologies. However, despite extensive research in this area, to our knowledge, there has been no investigation into the effects of incorporating logical constraints on the resilience of deep learning models against adversarial attacks. We hypothesise that by incorporating background knowledge into the deep learning framework, these algorithms can leverage human-like cognitive abilities, potentially boosting their robustness against adversarial attacks.

Moreover, studies typically rely on human expertise to derive logical constraints, which may not always yield sufficient or optimal rules. To address this, we propose utilizing ILP systems to automatically generate logical constraints.

This paper introduces an innovative method, RLDL, to enhance the trustworthiness of NN-based traffic sign recognition. Our approach integrates human knowledge to improve reliability, mirroring human perception. We automatically extract logical rules by leveraging ILP from a few positive and negative examples. These rules are subsequently mapped into a logical constraint matrix. We then evaluate the satisfaction of constraints for each prediction, and the resulting constraint dissatisfaction is integrated into the loss function, ensuring compliance with background knowledge.

II. Related Work

The field of artificial intelligence is actively investigating ways to incorporate prior knowledge into NNs to achieve various objectives, such as improving performance, learning from limited data, and ensuring compliance with the provided background knowledge [3]. This integration seeks to bring NNs closer to emulating human learning processes, where high-level features and their associated logical constraints play a crucial role in cognition.

This section provides an overview of current research efforts aimed at integrating background knowledge to enhance NN performance and capabilities. These approaches can be broadly categorised into two main classes: Loss Function Modification and Network Architecture Adjustment. Our approach falls into the former category, where constraints are infused into the NN via loss functions. This method has been explored in prior works such as [4]–[6], where the satisfaction of logical constraints in predictions is integrated into the loss function to enforce consistency with the desired outputs. In a similar way, Wang & Pan [7] took an approach by employing a parallel neuro-reasoning engine. This engine generates an output consistent with the neural process, and the disparity between the two outputs is subsequently incorporated into the loss function. This innovative method aims to leverage the strengths of both neural networks and reasoning engines to improve model performance and adherence to logical constraints.

[1]Zahra Chaghazardi is with the Dept. of Computer Science, University of Surrey, The United Kingdom, z.chaghazardi@surrey.ac.uk

[2]Saber Fallah is with Mechanical Engineering Sciences, University of Surrey, The United Kingdom, s.fallah@surrey.ac.uk

[3] Alireza Tamaddoni-Nezhad is with the Dept. of Computer Science, University of Surrey, The United Kingdom, a.tamaddoni-nezhad@surrey.ac.uk

Similarly, Wang and Pan [7] employed a parallel neuro-reasoning engine, generating an output consistent with the neural process and incorporating the disparity between the two outputs into the loss function. This method aims to leverage both neural networks and reasoning engines to enhance model performance and adherence to logical constraints.

In the second category, various approaches have been developed to integrate constraints into neural networks by adjusting their topology. One study in this category is MultiplexNet, where the prediction layer of the neural network is augmented with specific transformations, treating the deep network's output layer like a logical circuit multiplexor. This approach enables logical branching within the network architecture, ensuring full constraint satisfaction [8].

Coherent-by-Construction Network (CCN) approach [9] offers a dual methodology for integrating constraints into neural networks. This involves introducing a supplementary top layer that modifies the output to adhere to specified constraints and integrating constraints into a collaborative loss function. Ahmed et al. introduced the Semantic Probabilistic Layer (SPL) [10], a method incorporating a compiled logic circuit layer into the network to enforce constraints. Alternatively, Yang et al. introduced NeurASP [11], a framework designed to incorporate neural networks into answer set programs, establishing a correlation between logic predicates and their neural counterparts and Manhaeve et al. [12] introduced Probabilistic ILP which integrates rule-based learning with statistical learning and create a parallel between the logic predicates with neural predicates. For a comprehensive overview of incorporating logical constraints into deep learning, refer to [3].

III. Methodology

In this section, we present our methodology designed to enhance the robustness of Neural Networks against adversarial attacks and anomalies within the domain of Autonomous Vehicles (AV). Our approach incorporates supplementary attributes and integrates their logical constraints during the model training phase. Furthermore, our approach employs ILP to systematically extract rules and constraints from human-provided knowledge. Combining logical constraints to NNs, significantly contributes to the model's resilience, enhancing its performance in the challenging environment of AV applications.

The architecture of our proposed Neuro-symbolic traffic sign classifier is depicted in Fig. 1. This process involves two interconnected parts: Symbolic and Deep Learning. In the symbolic framework, initially, human knowledge is fed into an ILP system, where the extracted rules are subsequently mapped to logical constraints. These logical constraints are then fed into the loss calculator.

On the other hand, during the NN model's training process, an image is presented with three labels denoting the traffic sign class, shape, and colour. The Convolutional Neural Network (CNN) model processes this input and provides predictions for all three labels. These predictions are then

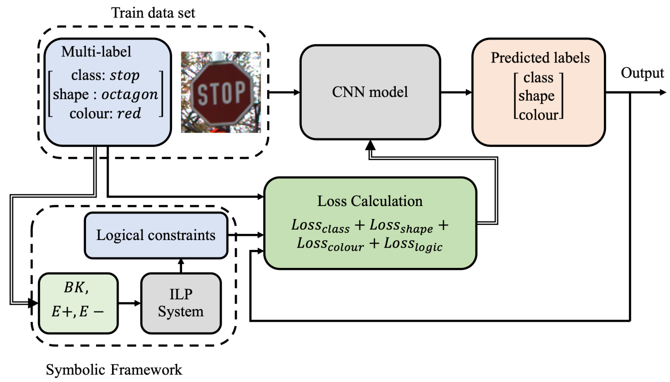


Fig. 1: Robust Logic-infused Deep Learning traffic sign classifier

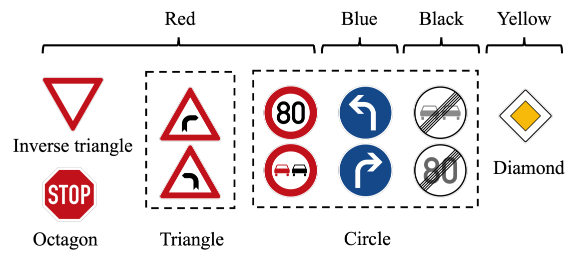


Fig. 2: A set of eleven traffic signs utilized in this study, each associated with corresponding shape and colour labels.

fed into the loss calculator, where the satisfaction of the predicted labels with logical rules is quantified. Subsequently, a regularization term is incorporated into the loss function, ensuring adherence to the provided rules.

In this classification task, it is important to note that the ILP system has learned one rule for each class. Consequently, each traffic sign is associated with only one specific rule, and throughout the training, the rule exhibiting the highest agreement with the prediction is selected to regulate the loss function.

A. Material and Method

We developed a multilabel classifier by adapting a well-known and publicly available implementation of CNN architecture [13] renowned for its high performance in traffic sign recognition. The customized model is depicted in Fig. 3 and serves as our baseline DL model.

The model was configured with the Adam optimizer (learning rate: 0.001), trained over 20 epochs with a batch size of 32, using Cross Entropy as the loss function.

This multilable classifier is trained on the German Traffic Sign Recognition Benchmark (GTSRB) [14]. Our training utilized a subset of the GTSRB dataset, comprising eleven distinct traffic signs each contains 150 training images. It is noteworthy that the overall training dataset comprised a total of 1650 images.

Fig. 2 depicts the eleven classes utilized in our study along with their respective categories, categorized based on colour and shape.

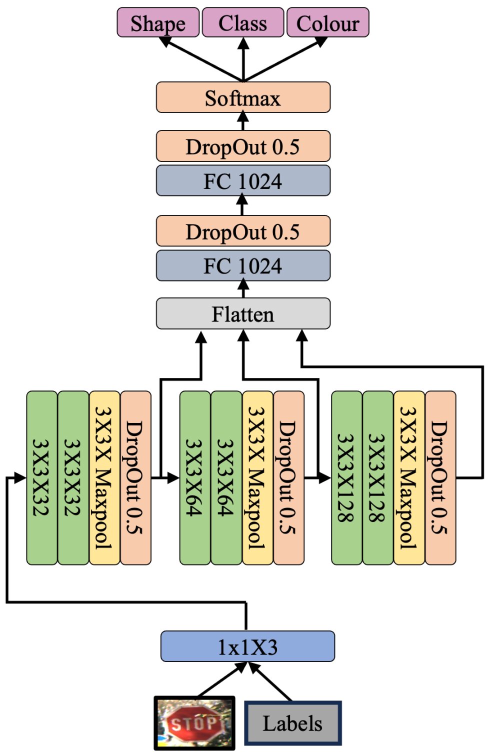


Fig. 3: Architecture of the CNN Model Utilized as a baseline in our study.

We employed the same approach in the testing phase, selecting 50 images from each class within the GTSRB dataset. Additionally, we assessed the classifiers' robustness against adversarial attacks, employing various test datasets subjected to distinct attack methodologies.

In this study, the proposed classifier undergo testing across various adversarial attack datasets, including Subtle and LoveHate datasets generated by the RP_2 attack, as proposed by Eykholt et al. [1], GRATS (Dirty) [15], Dart [16], and Shadow [17] attack datasets. Each test dataset presents distinct challenges, providing a comprehensive assessment of the classifiers' resilience under varied adversarial conditions.

B. Extracting Constraints

Deriving logical constraints directly from human experts may not always result in comprehensive and optimal rules, leading to concerns about the reliability and comprehensiveness of such constraints. In response to this challenge, we propose adopting ILP systems to extract logical constraints automatically.

ILP is a predominant approach that combines machine learning with formal logic and is designed to derive logical rules from a limited set of examples, as proposed by Muggleton [18]. The ILP system employed in this framework is Aleph [19], and the corresponding algorithm is illustrated in Algorithm. 1.

The algorithm iteratively selects positive examples, generates bottom clauses based on language bias, and adds them to the hypothesis set while pruning covered examples. The

Algorithm 1 ILP algorithm

Input: BK, E^+, E^-, Language biase $\mathcal{L}$
Output: hypothesis H

$H \leftarrow \emptyset$
while $E^+ \neq \emptyset$ **do**
  $e \leftarrow select(E^+)$
  $\perp_e \leftarrow bottom\_clause(e, \mathcal{L})$
  $c \leftarrow clause\_reduction(\perp_e)$
  $H \leftarrow H \cup c$
  $E^+ \leftarrow prune(E^+)$: remove E^+ covered by c
end while
return H

process continues until there are no more positive examples. The result is a set of induced rules or hypotheses that should cover as many positive and as few negative examples as possible. Background knowledge (BK) provided by humans, should include all essential predicates to represent the relevant information for inducing the rules.

This framework comprises three phases. In the first phase, it takes BK, positive examples E^+, and negative examples E^-. To illustrate this process, consider the following example: Suppose we have background knowledge stating $has\_shape(A, circle)$, $has\_colour(A, red)$, $has\_shape(B, circle)$ and $has\_colour(B, black)$, providing information about A and B traffic signs. Additionally, we have a positive example, $traffic\_sign(A, no\_passing\_sign)$. Also, there are negative examples representing traffic signs which are not $No\ Passing$ signs such as $traffic\_sign(B, no\_passing\_sign)$. In the second phase, ILP will drive this rule:

```
traffic_sign(A, no_passing_sign):-
        has_shape(A, circle),
        has_colour(A, red).
```

This rule implies that sample A represents a 'No Passing' traffic sign if its shape is a circle and its colour is red.

In the subsequent phase, after the ILP system derives interpretable logical rules, the third stage involves mapping these rules to the creation of a constraint matrix denoted as C represented as:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1L} \\ c_{21} & c_{22} & \cdots & c_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NL} \end{bmatrix}$$

In this context, N represents the number of rules or constraints, while L denotes the number of attributes, such as shape or colour. Each attribute is characterized by distinct indices that signify its specific category. For instance, within the 'shape' attribute, indices may correspond to shapes like circles, triangles, and so forth. This matrix structure captures the relationships between rules and attributes, providing a systematic representation of logical constraints within our system.

Algorithm 2 RLDL Logical Loss

Input:

- Logical Constraint (C) from ILP
- Predicted feature vectors ($P\_class, P\_shape, P\_colour$) from DL

Output: Logical Loss (L)

$final\_satisfaction \leftarrow []$
for each rule $c \in C$ **do**
  $class\_id, shape\_id, color\_id \leftarrow c$
  $p\_class\_c \leftarrow P\_class[class\_id]$
  $p\_shape\_c \leftarrow P\_shape[shape\_id]$
  $p\_color\_c \leftarrow P\_colour[colour\_id]$
  if Logic == 'product' **then**
    $satisfaction \leftarrow p\_class\_c \times p\_shape\_c \times p\_colour\_c$
    $final\_satisfaction \leftarrow final\_satisfaction \cup satisfaction$
  else if Logic == 'Gödel' **then**
    $satisfaction \leftarrow \min(p\_lass\_c, p\_shape\_c, p\_color\_c)$
    $final\_satisfaction \leftarrow final\_satisfaction \cup satisfaction$
  end if
end for
$max\_satisfaction \leftarrow \max(final\_satisfaction)$
$Loss \leftarrow 1 - max\_satisfaction$
return $Loss$

Additionally, c_{ij} represents the value of the index related to attribute j, associated with constraint i. Each rule is expressed as a logical combination of L attributes where each attribute l includes S_l categories. In this setup, the size of the matrix will be $N \times L$, which is significantly smaller than the proposed constraint matrix by [5], having the dimension of $N \times (S_1 + \ldots + S_L)$.

Considering our case study with 11 rules, 4 colours, 5 shapes, and 11 classes, the constraint dimension for our proposed method is 11×3, while for the previous method, it would be $11 \times (11 + 4 + 5)$. This difference, particularly for larger problems, may lead to efficiency issues, highlighting the advantage of our approach.

C. Loss function

In the training process, we adopt the multi-label cross-entropy loss function. To enhance the robustness of our models, we introduce a regularization term crafted to incorporate the level of satisfaction with the applied logical constraints. The regularization process contributes to their ability to adhere to logical constraints and improve overall performance.

The algorithm presented in Algorithm. 2 demonstrates the process of computing the loss function. In order to quantify this regulatory aspect, we systematically evaluate the conformity of each prediction to a range of rules and identify the rule with the highest satisfaction.

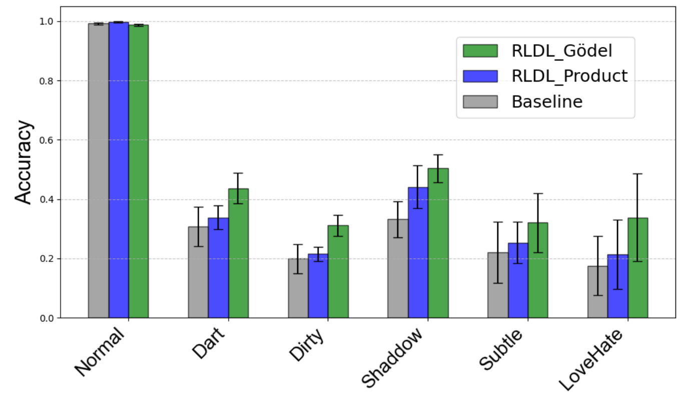


Fig. 4: Comparison of accuracy among baseline (described in Fig. 3) and logic-based models including RLDL_Product, and RLDL_Gödel on normal and targeted stop signs by various attack datasets including Dart, Dirty, Shadow, Subtle, and LoveHate.

Eq. 1 represents the total loss L_t, which is composed of two main components. The first component is the summation of cross-entropy losses associated with each attribute. Each L_l represents the loss associated with predicting the l-th attribute.

The second component, denoted by L_{logic}, represents an additional loss term related to logical constraints, which captures any deviations from the specified rules or constraints.

$$\mathrm{L}_t = \sum_{l=1}^{L} L_l + L_{logic} \tag{1}$$

Consider the rule matrix C of size $N \times L$, where each element C_{nl} signifies the indices of the attribute l associated with the respective rule n. For each attribute l, the deep learning model's output is encapsulated in a prediction vector P_l, containing l_S predictions, where S represents the number of categories within that attribute.

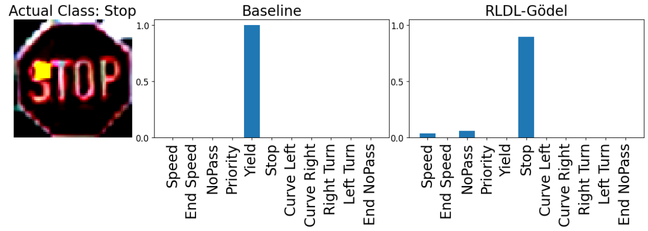
Given C and $P = P_1, ..., P_L$, our goal is to compute the degree of satisfaction of each constraint for each output. To achieve this, we introduce a matrix G of dimensions $N \times L$, each element G_{nl} is determined by the probability from prediction vectors $P_l(C_{nl})$. This relationship is expressed as:

$$G_{nl} = P_l(C_{nl}), \quad \text{for } n = 1, \ldots, N \text{ and } l = 1, \ldots, L \tag{2}$$

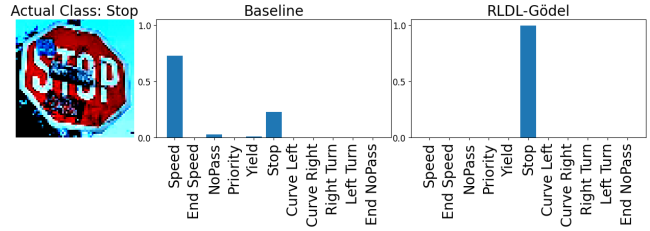
Subsequently, we compute the degree of satisfaction G' using a t-norm (conjunction) operation along the dimension two (attributes) followed by a t-conorm (disjunction) across the other dimension (rules):

$$G' = t - conorm(t - norm(G, dim = 2), dim = 1) \tag{3}$$

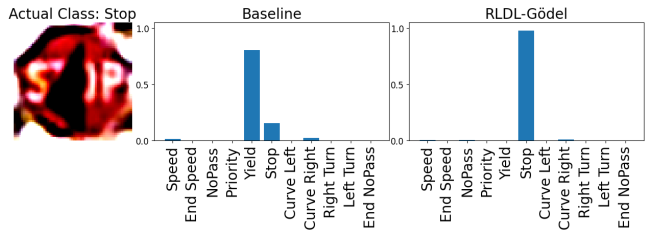
This expression utilises a t-norm operation to compute the satisfaction level for each rule. For instance, if we choose the Gödel t-norm, the calculation would be represented as $t\text{-norm}(G, \text{dim} = 2) = \min(G_{n,1}, \ldots, G_{n,l})$. Subsequently, it employs a t-conorm operation to identify the rule with the



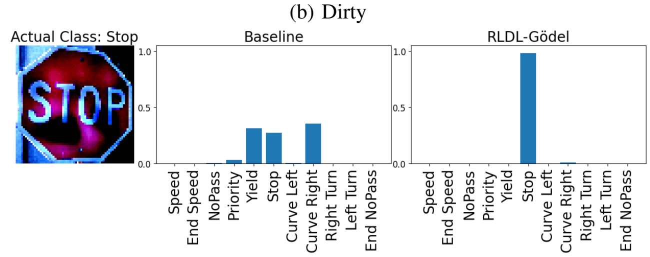
(a) Dart



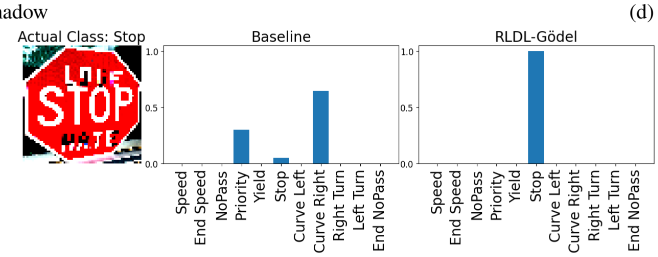
(b) Dirty



(c) Shadow



(d) Subtle



(e) LoveHate

Fig. 5: Exemplifying predictions generated by both the baseline and the proposed RLDL_Gödel models for targeted stop signs subjected to various attacks.

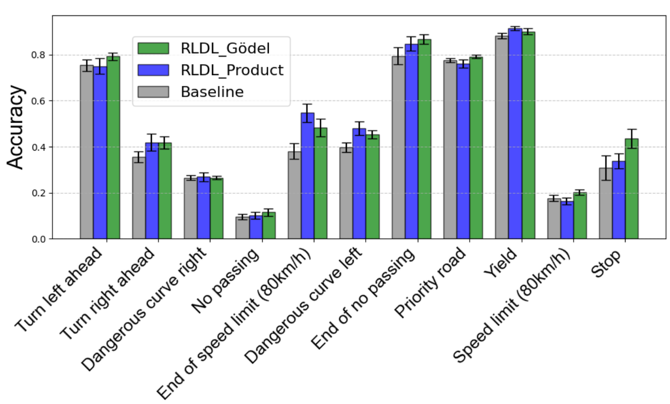


Fig. 6: Comparison of accuracy between baseline (pure deep learning) and logic-based models including RLDL_Product, and RLDL_Gödel across various traffic signs in the Dart adversarial attack dataset.

maximum satisfaction for prediction. Next, we define the logic loss L_{logic} as:

$$L_{logic} = 1 - G' \tag{4}$$

In this classification problem, only one rule should be correct for each instance. Subsequently, the dissatisfaction is computed and incorporated into the loss function as a regularisation term.

Specifically, we consider the Gödel and Product t-norms for evaluating the satisfaction of each rule. The logical conjunction operator is substituted with the product t-norm ($x \wedge_p y \equiv xy$) or the Gödel t-norm ($x \wedge_g y \equiv \min(x, y)$). Furthermore, the Gödel t-conorm is employed as a disjunction ($x \vee_g y \equiv \max(x, y)$). Let's consider an example with the following rule:

$$\text{Rule 1: no\_passing\_sign} \leftarrow \text{red} \wedge \text{circle}$$

Considering rule one, the system selects the probabilities associated with the attributes shape, colour and class as circle, red, and no passing sign. Suppose we have the following shape predictions for an input image:

$$\text{P\_shape} : [0.0, 0.3, 0.0, 0.7, 0.0]$$

This prediction implies a probability of 0.7 for the shape being a circle. Similarly, from P_colour and P_class, the deep learning model obtains probabilities of 0.9 for the colour being red and 0.8 for the sign being a no-passing sign.

According to the product t-norm, the rule one satisfaction is calculated by multiplying these probabilities together while Gödel t-norm selects the minimum value of them:

$$\text{Product t-norm rule 1 satisfaction} = 0.7 \times 0.8 \times 0.9 = 0.5$$

$$\text{Gödel t-norm rule 1 satisfaction} = \min(0.7, 0.8, 0.9) = 0.7$$

This process is repeated for all rules, and the rule with the maximum satisfaction is selected as the best choice. The complement of this satisfaction value (0.5 and 0.3 for product and Gödel t-norm, respectively) is then added to the loss function as a regularisation term. This mechanism ensures that the model adheres to the imposed logical constraints while enhancing overall performance.

IV. Results and Discussion

Each model underwent ten training iterations during the training process, and the average of accuracies was computed. As shown in Fig. 4, the accuracy achieved by three different models: baseline, RLDL_Product, and RLDL_Gödel, when tested on targeted stop signs. The testing is conducted across various datasets, including Normal, Dart, Dirty, Shadow, Subtle, and LoveHate.

The results indicate that while RLDL_Gödel marginally outperforms the baseline model on the Normal dataset, it significantly surpasses other models' accuracy when subjected to attack datasets. This underscores the robustness conferred by RLDL integration. Although RLDL_Product also outperforms the baseline model in all attack methods, its performance consistently lags behind RLDL_Gödel, highlighting the latter's superior robustness against adversarial attacks.

Fig. 5 showcases predictions generated by both the baseline and proposed RLDL_Gödel models for targeted stop signs under different attack scenarios, including Dart, Dirty, Shadow, Subtle, and LoveHate attacks. The results indicate that the RLDL_Gödel model produces more robust predictions with high confidence compared to the baseline model predictions.

Fig. 6 depicts the accuracy comparison among the baseline model, RLDL_Product, and RLDL_Gödel logic-based models across various traffic signs within the Dart dataset. Overall, the RLDL models demonstrate improved accuracy across the majority of traffic signs.

V. Conclusion

In summary, our experiments demonstrate the significant advancements of the proposed RLDL model, particularly in accurately detecting adversarial traffic signs. Despite training on a relatively limited dataset of 1650 images, the RLDL model exhibits substantial enhancements in accuracy, underscoring its robustness in handling challenging scenarios.

A key aspect of our approach is the integration of logical rules into the neural network architecture. This fusion enhances the model's performanceand fortifies its reliability. By incorporating domain-specific knowledge through logical rules, the RLDL model can better generalize from limited data and correctly recognise traffic signs with adversarial manipulations that might deceive traditional neural networks.

These findings underscore the potential of the RLDL approach as a valuable strategy for bolstering the trustworthiness of neural networks in real-world applications, particularly in the domain of autonomous vehicles. By combining the strengths of deep learning and symbolic logic, the RLDL model presents a robust framework for enhancing the accuracy and reliability of vision systems in autonomous driving. Future work could focus on expanding the dataset and refining logical rules to further boost the model's performance across various scenarios.

Acknowledgment

The first author would like to acknowledge her PhD grant funding from the Breaking Barriers Studentship Award at the University of Surrey.

References

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[2] Z. Chaghazardi, S. Fallah, *et al.*, "A logic-based compositional generalisation approach for robust traffic sign detection," in *International Joint Conference on Artificial Intelligence 2023 Workshop on Knowledge-Based Compositional Generalization*, 2023.

[3] E. Giunchiglia, M. C. Stoian, and T. Lukasiewicz, "Deep learning with logical constraints," *arXiv preprint arXiv:2205.00523*, 2022.

[4] M. Diligenti, S. Roychowdhury, and M. Gori, "Integrating prior knowledge into deep learning," in *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2017, pp. 920–923.

[5] E. Giunchiglia, M. C. Stoian, S. Khan, F. Cuzzolin, and T. Lukasiewicz, "Road-r: The autonomous driving dataset with logical requirements," *Machine Learning*, pp. 1–31, 2023.

[6] M. C. Stoian, E. Giunchiglia, and T. Lukasiewicz, "Exploiting t-norms for deep learning in autonomous driving," *arXiv preprint arXiv:2402.11362*, 2024.

[7] W. Wang and S. J. Pan, "Integrating deep learning with logic fusion for information extraction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 9225–9232.

[8] N. Hoernle, R. M. Karampatsis, V. Belle, and K. Gal, "Multiplexnet: Towards fully satisfied logical constraints in neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 5, 2022, pp. 5700–5709.

[9] E. Giunchiglia and T. Lukasiewicz, "Multi-label classification neural networks with hard logical constraints," *Journal of Artificial Intelligence Research*, vol. 72, pp. 759–818, 2021.

[10] K. Ahmed, S. Teso, K.-W. Chang, G. Van den Broeck, and A. Vergari, "Semantic probabilistic layers for neuro-symbolic learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 944–29 959, 2022.

[11] Z. Yang, A. Ishay, and J. Lee, "Neurasp: Embracing neural networks into answer set programming," *arXiv preprint arXiv:2307.07700*, 2023.

[12] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "Deepproblog: Neural probabilistic logic programming," *Advances in neural information processing systems*, vol. 31, 2018.

[13] Vivek Yadav, "German sign classification using deep learning neural networks," https://github.com/vxy10/p2-TrafficSigns, 2016.

[14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

[15] K. D. Apostolidis, E. V. Gkouvrikos, E. Vrochidou, and G. A. Papakostas, "Traffic sign recognition robustness in autonomous vehicles under physical adversarial attacks," in *Cutting Edge Applications of Computational Intelligence Tools and Techniques*. Springer, 2023, pp. 287–304.

[16] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "Darts: Deceiving autonomous cars with toxic signs," *arXiv preprint arXiv:1802.06430*, 2018.

[17] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 345–15 354.

[18] S. Muggleton, "Learning from positive data," in *International conference on inductive logic programming*. Springer, 1996, pp. 358–376.

[19] Ashwin Srinivasan, "The aleph manual," https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html, 2001.