Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

NeuroShield: A Neuro-Symbolic Framework for Adversarial Robustness

Ali Shafiee Sarvestani
University of Illinois at Chicago
Jason Matthew Schmidt
University of Illinois at Chicago
Arman Roohi
aroohi@uic.edu

University of Illinois at Chicago

Article

Keywords:

Posted Date: May 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8282780/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

NeuroShield: A Neuro-Symbolic Framework for Adversarial Robustness

Ali Shafiee Sarvestani[1], **Jason Schmidt**[2], **and Arman Roohi**[1*]

[1]University of Illinois Chicago, Department of Electrical and Computer Engineering, Chicago, IL, 60607, USA
[2]University of Illinois Chicago, Department of Computer Science, Chicago, IL, 60607, USA
[*]aroohi@uic.edu

ABSTRACT

Adversarial vulnerability and lack of interpretability are critical limitations of deep neural networks, especially in safety-sensitive settings such as autonomous driving. We introduce **`NeuroShield`**, a neuro-symbolic framework that integrates symbolic rule supervision into neural networks to enhance both adversarial robustness and explainability. Domain knowledge is encoded as logical constraints over appearance attributes such as shape and color, and enforced through semantic and symbolic logic losses applied during training. Using the GTSRB dataset, we evaluate robustness against FGSM and PGD attacks at a standard ℓ_∞ perturbation budget of $\varepsilon = 8/255$. Relative to clean training, standard adversarial training provides modest improvements in robustness ($\sim$10 percentage points). Conversely, our FGSM-Neuro-Symbolic and PGD-Neuro-Symbolic models achieve substantially larger gains, improving adversarial accuracy by 18.1% and 17.35% over their corresponding adversarial-training baselines, representing roughly a three-fold larger robustness gain than standard adversarial training provides when both are measured relative to the same clean-training baseline, without reducing clean-sample accuracy. Compared to transformer-based defenses such as LNL-MoEx, which require heavy architectures and extensive data augmentation, our PGD-Neuro-Symbolic variant attains comparable or superior robustness using a ResNet18 backbone trained for 10 epochs. These results show that symbolic reasoning offers an effective path to robust and interpretable AI.

Introduction

The black-box nature of deep learning models makes them vulnerable to adversarial attacks and failure modes. Their lack of transparency limits human interpretability and supervision. Traditional human oversight is primarily limited to dataset preparation and pre-processing, which is insufficient for safety-sensitive tasks requiring trust and explainability.

Neuro-Symbolic AI (NSAI) augments deep neural networks (DNNs) with symbolic reasoning to enhance interpretability, trust, and robustness[1]. DNNs can extract abstract features from large datasets and learn complex decision boundaries[2]. However, their inner mechanisms remain opaque, forcing users to trust predictions without meaningful insight into the decision-making process. Such opacity can cause harmful errors in real-world deployments. NSAI addresses this by incorporating domain-specific symbolic rules into neural models. These symbolic components act as expert supervisors, verifying whether the model's output adheres to known logical relationships. For example, in the classification of traffic signs, a symbolic rule might require that a "STOP" sign be red and octagonal. If a model predicts a class without detecting these attributes, the symbolic module can flag the inconsistency, Fig.1. Depending on the integration strategy, the symbolic component may function as a post-hoc verifier, a parallel logic engine, or an embedded structure that influences training dynamics. Some approaches integrate logic into the loss; others use symbolic rules to filter or re-rank outputs[3,4]. NSAI systems may rely on predefined rules, analogous to supervised learning, or dynamically extract new rules during training through data-driven induction mechanisms. Despite strong performance, DNNs remain highly susceptible to adversarial threats. These threats arise from small, intentional perturbations, often imperceptible to the human eye, designed to mislead the model. This vulnerability stems from reliance on superficial statistical correlations rather than semantic reasoning. Thus, even well-trained models can be fooled by inputs that look nearly identical to clean samples. Adversarial attacks can occur during both training and inference. Some corrupt training by injecting malicious samples; others target inference by inducing confident but incorrect predictions[5]. These attacks may be designed to be broad and random, or highly targeted toward specific outputs. In either case, they exploit the lack of reasoning and interpretability, revealing the brittleness of neural models.

In response, various defense strategies have been developed to make the models more resilient. These generally involve either adjusting the training process to expose the model to noisy or perturbed input, modifying inputs at test time to suppress adversarial patterns, or altering the model architecture to obscure the gradients that attackers rely on. Some approaches attempt to guarantee robustness through formal verification techniques, while others aim to reduce the model's sensitivity to input changes[5]. However, all of these approaches operate purely in the neural domain and remain limited by training data. This

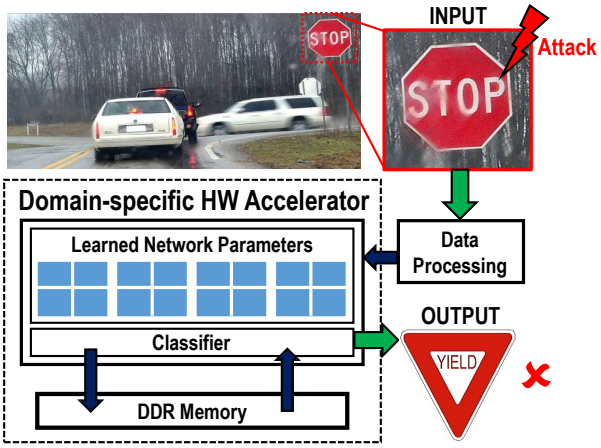

Figure 1. Soft errors in an NN accelerator can lead to image misclassification in a self-driving car, potentially causing it to accelerate instead of braking abruptly. (adopted from[6]).

limitation motivates integrating symbolic reasoning into learning. Unlike purely statistical models, neuro-symbolic systems can evaluate predictions through the lens of logic and structured knowledge. When encountering an adversarial input, such a system is not solely dependent on learned patterns; it can also rely on logical constraints representing domain knowledge. These constraints help ensure that output remains consistent with fundamental relationships, such as a known traffic sign's shape and color requirements. Even if a perturbed input misleads the neural component, the symbolic module acts as a semantic filter, identifying contradictions. In this way, reasoning capabilities offer an additional layer of defense, one that enhances robustness by establishing predictions in interpretable and verifiable rules rather than purely statistical features.

This paper presents **`NeuroShield`**, an NSAI framework designed to improve the explainability and robustness of neural networks by integrating symbolic reasoning into the learning pipeline. The remainder of this paper is organized as follows. Section 1 reviews related work on neuro-symbolic learning and adversarial robustness. Section 2 presents our rule-based supervision framework. Sections 3 and 4 detail the backbone modifications and the training pipeline. Section 6 evaluates the robustness, interpretability, and efficiency of our model.

1 Related Works

Since our method integrates symbolic reasoning into neural networks for adversarial robustness, we organize the related work into two parts: symbolic reasoning, which reviews neuro-symbolic frameworks that embed logic into neural learning, and adversarial defenses, which survey techniques for defending against adversarial attacks.

1.1 Symbolic Reasoning

Recent advances in neuro-symbolic AI have explored integrating symbolic reasoning with neural network learning to improve interpretability, consistency, and generalization. Logic Tensor Networks (LTNs)[3] combine first-order logic with deep learning by embedding logical symbols (constants, functions, predicates) into real-valued tensors through a differentiable logic system called Real Logic. LTNs support fuzzy truth values, logical operations, and quantifiers, and have been applied to classification, regression, and semi-supervised tasks. Neural Logic Machines (NLMs)[7] extend this idea with a differentiable architecture for relational reasoning, using weight sharing and pooling to learn logic-like rules and generalize across tasks such as transitive reasoning and program induction.

Neuro-Symbolic Forward Reasoning (NSFR)[8] integrates symbolic logic with perception by extracting object-centric representations from images and mapping them to logical atoms through predicate classifiers. These atoms are processed by a differentiable forward-chaining engine that performs multi-step inference over fuzzy truth values, achieving interpretable results on visual reasoning tasks such as Kandinsky Patterns and CLEVR-Hans. Unlike methods that rely solely on latent features, NSFR emphasizes rule-driven behavior grounded in perception. Semantic loss[9] provides another way to embed symbolic constraints by defining a differentiable loss over propositional logic formulas. It penalizes confidence in logic-violating outputs and has been effective in semi-supervised classification and structured prediction tasks such as path generation and preference ranking.

Finally, neuro-symbolic verification[10] applies symbolic reasoning to model verification by combining neural networks with logic-based assertion checking. A specification network defines semantic conditions, and verification rules are encoded in a

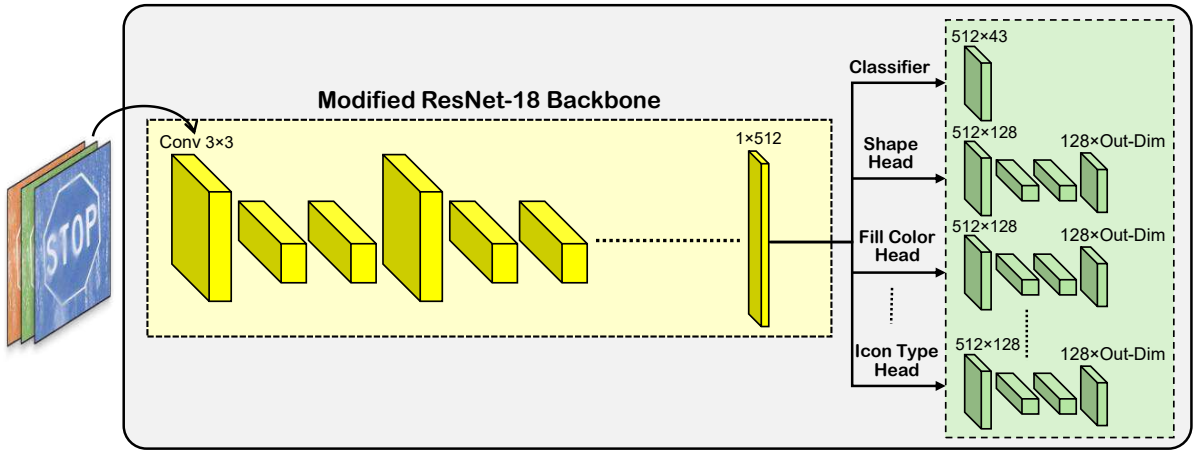

Figure 2. Modified ResNet18 architecture used in our framework. The network preserves high spatial resolution in early layers by replacing the initial 7×7 convolution with a 3×3 kernel and removing the max pooling layer. The final fully connected layer is removed, and the resulting 512-dimensional feature vector is shared between the traffic sign classifier and symbolic attribute heads.

formal assertion language and checked using SMT (Satisfiability Modulo Theories) solvers, enabling validation of semantic and multi-network properties for safety-critical systems.

1.2 Adversarial Robustness

In this section, we review defense strategies designed to improve robustness against adversarial attacks, including adversarial training, input transformation, adversarial detection, and certified defenses. Recent studies show that adversarial training remains one of the most effective defense mechanisms. Fast Gradient Sign Method (FGSM) introduced in[11], generates adversarial examples via a single gradient-sign step and showed substantial robustness gains on MNIST, recovering over 60 percentage points of adversarial accuracy while maintaining high clean accuracy. However, FGSM training may cause gradient masking and fails against stronger iterative attacks. To overcome these limitations, in[12], adversarial training was formalized as a robust optimization problem, and Projected Gradient Descent (PGD) was introduced as a universal first-order adversary. PGD-based adversarial training demonstrated significantly improved robustness, achieving up to 89% precision under the PGD attack on MNIST, and became the de facto baseline for adversarial defenses. Nonetheless, PGD training is computationally intensive and can overfit the specific attack used during training, reducing its effectiveness against unseen or adaptive threats.

Beyond adversarial training, input transformation is a practical defense that attempts to remove perturbations before classification. Common techniques include JPEG compression, bit-depth reduction, pixel dropout, and total variation minimization. These methods are model-agnostic and easy to apply, but often reduce clean accuracy and remain vulnerable to adaptive attacks. Many input-based defenses have been shown to rely on gradient obfuscation, which can give a false sense of security and be circumvented by carefully designed attacks[13]. Another defense direction is adversarial example detection, which uses auxiliary models or statistical tests to identify adversarial inputs by examining feature activations, gradients, or output distributions. While sometimes effective, detection approaches often fail to generalize across datasets or attacks and are particularly weak in white-box settings where the detector can be directly targeted[14]. Lastly, certified defenses aim to provide formal robustness guarantees by designing models that are provably resistant to perturbations within a given norm ball. Techniques such as randomized smoothing, Lipschitz regularization, and convex relaxation fall into this category. Although these approaches offer provable robustness, they are computationally expensive and struggle to scale to large models and complex datasets[15].

2 Rule Representation and Symbolic Supervision

Herein, we represent prior knowledge using symbolic rules that associate each traffic sign class with its expected attributes, including shape, fill and border color, icon or digit type, and category. These rules form a lightweight knowledge base that enables the model to interpret what a prediction semantically entails. For instance, a STOP sign should be red, octagonal, contain white text with a central "STOP" icon, have a white border, and belong to the priority category. These associations are defined statically using class-wise dictionaries (e.g., `class_shape_map`, `class_color_parts`) that encode expected values for each symbolic property. Each rule is translated into a soft implication of the form:

$$\text{if class} = c \Rightarrow \text{shape} = s,\ \text{color(fill)} = f,\ \text{category} = k,\ \ldots \tag{1}$$

We incorporate two forms of symbolic regularization to improve both interpretability and robustness. First, inspired by semantic loss[9], we define a simplified variant that encourages the predicted class distribution to align with a group of symbolically equivalent classes. Instead of forcing the model to always predict one exact label, this loss allows it to distribute confidence across all symbolically valid classes (e.g., all speed limit signs). This makes the model's mistakes more interpretable. Given the ground-truth label y, we identify all classes that share the same symbolic attributes, such as shape, color, icon type, and category, and construct a soft target distribution q:

$$q_i = \begin{cases} \frac{1}{|\mathscr{S}(y)|} & \text{if } i \in \mathscr{S}(y) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here, $\mathscr{S}(y)$ denotes the set of symbolically equivalent classes for the label y, and $q \in \mathbb{R}^{43}$ defines the soft target distribution over classes.

The semantic loss is computed as the Kullback–Leibler divergence between this symbolic target q and the predicted class distribution p_{cls}:

$$\mathscr{L}_{\text{semantic}} = \text{KL}(q \parallel p_{\text{cls}}) \tag{3}$$

Kullback-Leibler (KL) divergence is a measure of how one probability distribution diverges from a second expected distribution. In this context, it penalizes the model when it assigns high confidence to classes that are not symbolically consistent with the true label.

Second, we introduce a symbolic logic loss that enforces consistency between the predicted class and its associated symbolic components. Rather than enforcing hard restrictions, we use soft implications inspired by the Pylon framework[16]. For each component, we compute a soft truth value:

$$T(A \Rightarrow B) = \begin{cases} \frac{p_B}{p_A + \varepsilon} & \text{if } p_A > p_B \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Here, p_A is the confidence in the predicted class, and p_B is the confidence in the associated symbolic attribute. $T(A \Rightarrow B) \in [0,1]$ represents the degree to which the symbolic implication is satisfied. This formulation ensures that when the model confidently predicts a class (i.e., p_A is high), it is also held responsible for assigning high confidence to the corresponding symbolic attributes (i.e., p_B). If p_B is significantly lower than p_A, the implication truth drops, signaling a violation and resulting in a higher loss. In contrast, if p_A is low or $p_B \geq p_A$, the implication is considered satisfied and no penalty is applied. This soft implication mechanism, as defined in Equation (4), enforces the principle that confident predictions must be semantically coherent. The resulting symbolic logic loss penalizes these violations using a logarithmic function.

$$\mathscr{L}_{\text{logic}} = -\log(T(A \Rightarrow B) + \varepsilon) \tag{5}$$

In preliminary experiments, this independent logic consistency loss had only a limited impact on robustness, largely because each symbolic attribute was enforced in isolation. This led us to introduce a joint rule loss that treats the full set of symbolic attributes for a class as a single structured constraint, penalizing predictions whose combined attributes do not support the predicted label. Rather than checking shape, color, category, and icon type separately, the joint loss evaluates whether the entire attribute configuration is logically consistent with the class. In this way, it extends the basic logic loss from individual soft implications to a unified multi-attribute consistency constraint. The joint symbolic logic loss is defined as:

$$\begin{aligned} \mathscr{L}_{\text{joint}} &= -\log(T_{\text{joint}} + \varepsilon) \\ T_{\text{joint}} &= \min\left(\frac{\prod_j p_j^{w_j}}{p_{\text{cls}} + \varepsilon},\ 1\right) \end{aligned} \tag{6}$$

Here, p_j is the predicted probability for the symbolic attribute j, and w_j is a tunable importance weight. The numerator captures weighted symbolic consistency, and the denominator ensures normalization relative to class confidence. To make logic loss more adaptive, we update the weights w_j in Equation (6) at each epoch based on the symbolic accuracy of the model. Components with lower accuracy (e.g., shape, text, or icon) receive higher weights, which encourages the model to focus more on correcting those inconsistencies during training.

Furthermore, through these adaptive weights w_j, since the loss automatically allocates more pressure to whichever symbolic components are currently weakest, this encourages the backbone to organize its features so that each traffic-sign class occupies

a distinct, rule-consistent region in the shared representation space rather than relying on brittle, low-level cues. Finally, we combine symbolic regularization with standard supervised learning using total loss.

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{CE}} + \lambda_{\text{logic}} \cdot \mathscr{L}_{\text{logic}} + \lambda_{\text{semantic}} \cdot \mathscr{L}_{\text{semantic}} \tag{7}$$

where $\mathscr{L}_{\text{CE}}$ is classification cross-entropy, λ_{logic} and $\lambda_{\text{semantic}}$ are hyperparameters controlling the influence of logic and semantic consistency terms. In our formulation, the logic term L_{logic} corresponds to the joint symbolic loss defined in Equation (6). We first introduce the basic soft implication loss in Equation (5) for clarity, and then extend it into the unified joint symbolic loss in Equation (6), which is the version used during training. By integrating symbolic rules through soft constraints and semantic regularization, our framework improves both interpretability and adversarial robustness.

3 Backbone Modification and Symbolic Head Design

To enable simultaneous prediction of traffic sign classes and their symbolic attributes (such as shape, color, and icon type), we adapt the standard ResNet18 architecture as a shared backbone for both semantic classification and rule-based supervision. Specifically, we make two structural changes to preserve spatial detail in the early feature maps: the initial 7×7 convolution is replaced with a 3×3 kernel, and the max pooling layer is removed. This modification reduces early downsampling and preserves a higher spatial resolution, an essential factor for recognizing fine-grained small traffic signs. We also remove the final fully connected (FC) classification layer and replace it with an identity mapping. As a result, ResNet produces a 512-dimensional feature vector from its final global average pooling layer. This vector, denoted as $\mathbf{f} \in \mathbb{R}^{512}$, serves as the input to all downstream heads. This shared feature vector $\mathbf{f}$ is passed simultaneously to:

- a standard classification head for traffic sign recognition, and
- multiple symbolic attribute heads, each implemented as a lightweight multi-layer perceptron (MLP).

Each symbolic MLP head consists of:

- A linear projection layer that reduces dimensionality ($512 \rightarrow 128$),
- Batch normalization,
- ReLU activation, and
- A final linear layer that maps to the output space of a specific symbolic attribute (e.g., number of shapes or color classes).

Formally, each symbolic head can be written as:

$$\text{MLP}_{\text{attr}}(\mathbf{f}) = W_2 \cdot \text{ReLU}(\text{BN}(W_1 \cdot \mathbf{f}))$$

where $W_1 \in \mathbb{R}^{128\times512}$, $W_2 \in \mathbb{R}^{d\times128}$, and d is the number of output classes for the attribute (e.g., 5 shapes or 6 colors). We emphasize that no symbolic features are manually engineered or fed as input. All representations are learned directly from the raw images. Symbolic supervision is introduced through auxiliary outputs and corresponding logic-based losses, while the model remains fully end-to-end and differentiable. Figure 2 illustrates the modified architecture. The shared modified ResNet backbone extracts a global feature vector, which is then used by parallel heads for class and symbolic predictions.

4 Training Strategy with Symbolic and Adversarial Supervision

We train our model on the full German Traffic Sign Recognition Benchmark (GTSRB)[17] dataset using both clean and adversarial examples, guided by a combination of standard classification objectives and symbolic rule supervision. Each input is processed through the modified ResNet18 backbone and encoded into a 512-dimensional feature vector, which is shared between a classifier head and multiple symbolic heads to predict attributes such as shape, color components, category, and icon type. To increase robustness and interpretability, we jointly train the model using clean and adversarial samples. For each batch, adversarial examples are generated on-the-fly using the FGSM or PGD with a perturbation strength of $\varepsilon = 8/255$. This value provides a balanced adversarial threat: large enough to meaningfully perturb the input while remaining visually imperceptible under normalized image scales. Figure 3 visualizes how adversarial perturbations of increasing ε distort the input image, while remaining largely imperceptible at lower budgets. The model is supervised using three complementary loss functions: cross-entropy for classification accuracy, semantic loss to enforce symbolic class equivalence, and logic consistency loss to enforce adherence to symbolic rules. These losses are applied to both clean and adversarial samples, each using the formulation defined in Equation (7).

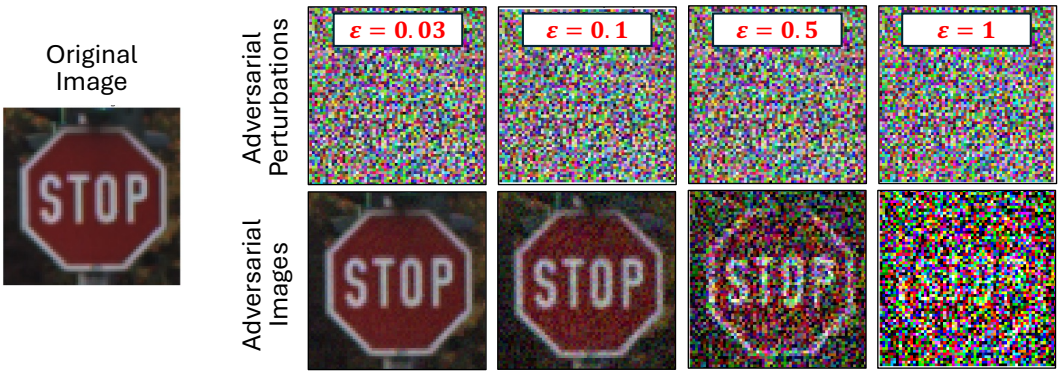

Figure 3. Original traffic sign image alongside adversarial perturbations and the corresponding adversarial examples generated at different perturbation budgets ($\varepsilon = 0.03$, 0.1, 0.5, and 1). The perturbation maps illustrate how increasing ε intensifies pixel-level noise, while the resulting adversarial images show progressively severe corruption of the underlying sign.

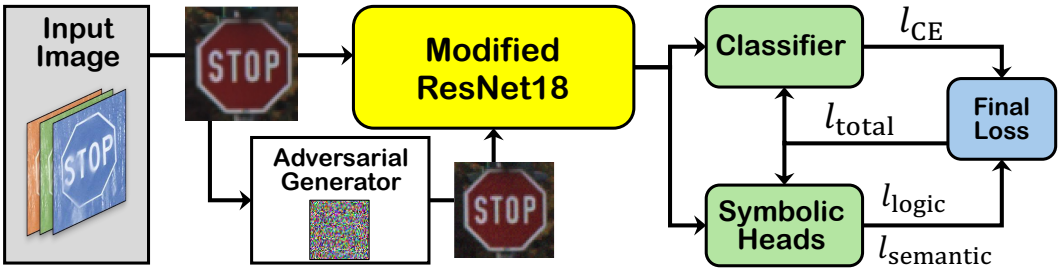

Figure 4. Illustration of our training strategy. Each input image is used to generate an adversarial example, and both clean and adversarial samples are passed through a shared modified ResNet18 backbone followed by the classifier and symbolic heads. The final loss combines classification cross-entropy with semantic and symbolic-logic losses, which jointly update the model.

The joint rule loss plays a particularly important role once adversarial training is introduced. Standard FGSM/PGD training only constrains the class logits to be stable under perturbations, which still allows the backbone to route gradients through brittle, non-semantic features as long as the top-1 label remains correct. In our framework, every adversarial example is simultaneously evaluated by the classifier head and the symbolic MLP heads. An adversarial input that preserves the predicted class label but causes one or more symbolic attributes (e.g., shape or category) to take inconsistent values incurs a large joint logic loss. Because all symbolic heads share the same 512-dimensional backbone representation, the gradients from L_{joint} force the backbone to organize its features so that robustness is achieved through semantically consistent attribute configurations, not in spite of them. In other words, the joint rule loss does not by itself create robustness, but when combined with adversarial training it disciplines how robustness is expressed, steering the network toward decision regions where both class predictions and symbolic attributes remain jointly stable under attack.

This joint training scheme ensures that the model learns to classify correctly under both natural and adversarial conditions, while also preserving symbolic coherence in its predictions. Unlike conventional adversarial training that focuses purely on classification robustness, our approach additionally penalizes violations of symbolic logic, yielding improved structure-aware generalization. Figure 4 illustrates our training pipeline. Each input image is first passed to the adversarial generator to produce its adversarial counterpart. Both the clean and adversarial samples are then fed into the modified ResNet18 backbone, followed by lightweight MLP heads, which output the semantic and symbolic predictions. The model is supervised on clean and adversarial samples using classification, semantic, and symbolic logic losses, and the total loss is derived from the weighted sum of these components. This architecture ensures robustness and logical consistency under both standard and adversarial conditions. To focus on symbolic regularization where it is most needed, we compute an adaptive logic weight λ_{logic} for each batch, based on classification cross-entropy and prediction correctness. This increases logic pressure on uncertain or misclassified samples, while reducing it when the model is confident and accurate. In contrast, the semantic loss weight $\lambda_{\text{semantic}}$ remains fixed, as its regularization effect is globally beneficial regardless of the certainty of the prediction. Finally, while symbolic reasoning introduces additional computational cost compared to standard adversarial training, the resulting improvements in interpretability and adversarial robustness justify the overhead, especially in safety-critical applications like traffic sign recognition.

Table 1. Comparison of clean and adversarial accuracy across different training strategies.

Training Method	Clean Accuracy*	FGSM Accuracy*	PGD Accuracy*
No Adversarial Training	98.67%	36.7%	25.5%
FGSM Training Only	96.5%	45.0%	35.0%
PGD Training Only	4.5%	43.6%	38.65%
FGSM-Clean Training	96.40%	46.70%	34.10%
PGD-Clean Training	7.00%	44.3%	39.70%
Neuro-Symbolic (ours)	97.90%	38.30%	19.90%
FGSM-Neuro-Symboli‘c (ours)	98.20%	63.10%	42.68%
PGD-Neuro-Symbolic (ours)	98.5%	65.75%	56.00%

*__Clean Accuracy__ = no attack; __FGSM Accuracy__ = accuracy under the FGSM attack; __PGD Accuracy__ = accuracy under the PGD attack

5 Evaluation

The proposed **`NeuroShield`** is evaluated on the GTSRB dataset. The model is trained from scratch for only 10 epochs, without any pre-training. This deliberate limitation serves two purposes. First, GTSRB has highly structured visuals, which makes models converge quickly; training for too long can therefore lead to overfitting to low-level features. Second, our goal is to encourage the model to rely more on symbolic reasoning and structural supervision, rather than memorizing dataset-specific patterns. Despite this short training schedule, our model achieves strong generalization, demonstrating the effectiveness of symbolic regularization. We evaluated the performance of our model in terms of both adversarial robustness and inference time computation overhead. First, we compare it with standard adversarial training methods, showing that the integration of symbolic reasoning significantly improves robustness under FGSM and PGD attacks while maintaining clean accuracy. We also benchmark our approach against a recent transformer-based defense method, demonstrating competitive or superior accuracy with much lower training cost and better interpretability. Finally, we analyze the inference-time computation overhead of our neuro-symbolic model, highlighting a favorable trade-off between robustness and efficiency, and discuss strategies for further reducing computational complexity in resource-constrained environments.

5.1 Comparison Results

5.1.1 Ours versus Standard Adversarial Training

We evaluated the robustness of the model under three conditions: standard clean test inputs, adversarial examples generated using FGSM, and those generated by the PGD attack. For both attack methods, we use a perturbation strength of $\varepsilon = 8/255$, which controls the maximum allowable change to each input pixel. This value is chosen to meaningfully distort the input while remaining visually imperceptible, thereby posing a realistic adversarial threat. Table 1 summarizes the results in several training configurations. The first column of this table reports test accuracy on clean samples, while the second and third columns show robustness under FGSM and PGD attacks, respectively. Our neuro-symbolic model combined with adversarial training (last two rows) achieves the highest accuracy under adversarial examples, while still maintaining over 98% accuracy on clean (non-adversarial) inputs, comparable to the best nonadversarial baseline.

For example, compared to FGSM adversarial training (45.0% FGSM accuracy), FGSM-Neuro-Symbolic achieves 63.10%, an improvement of +18.1%. Similarly, PGD-Neuro-Symbolic improves PGD accuracy from 38.65% (PGD adversarial training) to 56.00%, a gain of +17.35%. This shows that symbolic rule supervision significantly enhances robustness beyond what standard adversarial training achieves. Our pure Neuro-Symbolic model (without any adversarial training) maintains high clean accuracy ($\sim$ 97.90%) but drops under attack, especially PGD ($\sim$ 19.90%). This validates that symbolic logic alone is not enough. However, when combined with adversarial objectives, the hybrid model outperforms all other strategies under attack, demonstrating the complementarity of logical reasoning and adversarial supervision.

Interestingly, PGD-Clean achieves a very low clean accuracy ($\sim$ 7.00%), likely due to the inclusion of strong adversarial signals that distort the clean data representation. Although it maintains reasonable PGD accuracy ($\sim$ 39.70%), it underperforms compared to our hybrid model. Figure 5(a) shows the distribution of clean and adversarial samples for our hybrid Neuro-Symbolic model. Adversarial samples remain more closely grouped with their clean counterparts, indicating that the reasoning module helps preserve semantic structure and supports robust, rule-consistent predictions even under attack. In contrast, Figure 5(b) visualizes the distribution for a baseline ResNet model. The adversarial perturbations disrupt the structure of the internal feature space, leading to poor clustering and significant overlap between clean and adversarial samples.

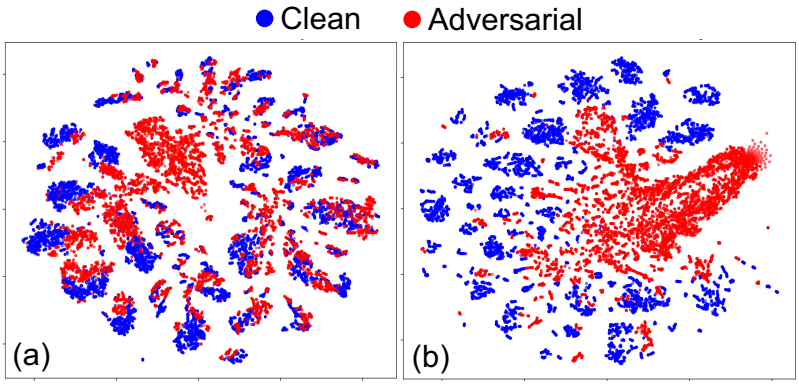

Figure 5. Comparison of clean and adversarial sample distributions using t-SNE projections. (a) Our Neuro-Symbolic model better preserves semantic structure, with adversarial samples remaining closely aligned with their clean counterparts, highlighting the benefits of rule-based reasoning in maintaining robustness. (b) Baseline ResNet model shows disrupted structure and significant overlap between clean and adversarial samples.

Table 2. Adversarial accuracy under FGSM and PGD attacks for different methods. Our neuro-symbolic variants are compared with transformer-based LNL[18] models with and without MoEx data augmentation.

Method	FGSM Accuracy	PGD Accuracy
LNL-Ti[18]	57.7%	37.9%
LNL-S[18]	64.5%	45.7%
LNL-MoEx-Ti[18] (Data Augmentation)	71.3%	55.4%
LNL-MoEx-S[18] (Data Augmentation)	77.8%	59.4%
FGSM-Neuro-Symbolic ($\varepsilon = 0.03$)	63.10%	42.68%
FGSM-Neuro-Symbolic ($\varepsilon = 0.5$)	74.00%	25.00%
PGD-Neuro-Symbolic ($\varepsilon = 0.03$)	65.75%	56.00%

5.1.2 Ours versus Transformer-Based Models

We further compare our neuro-symbolic framework with *LNL-MoEx*[18], a recent transformer-based defense method that integrates locality inductive bias with moment exchange (MoEx) to enhance adversarial robustness. LNL-MoEx is trained on GTSRB for 100 epochs and on CIFAR-10 for 150 epochs using a ViT-based architecture, and leverages MoEx as a form of statistical feature augmentation. Importantly, while many ViT-based defenses rely on ImageNet pre-training, the authors of LNL-MoEx emphasize that their method is trained from scratch on CIFAR-10 and GTSRB without ImageNet pre-training.

Table 2 presents a comparative analysis of adversarial accuracy under FGSM and PGD attacks. With $\varepsilon = 8/255 \sim 0.03$, our PGD-Neuro-Symbolic model achieves 56.00% PGD accuracy, slightly outperforming LNL-MoEx-Ti (55.4%) and approaching the best-performing transformer baseline, LNL-MoEx-S (59.4%), despite requiring no data augmentation and only 10 training epochs. Under FGSM attack with $\varepsilon = 8/255$, our model reaches 63.10% accuracy, which is competitive with LNL-S (64.5%) and significantly better than LNL-Ti (57.7%). Although LNL-MoEx-S achieves the highest FGSM accuracy (77.8%), it relies on transformer backbones and extensive data augmentation[18]. But the accuracy under PGD attack is more meaningful, because PGD performs multiple iterative gradient steps within the same ε-constraint, producing more effective and adversarially optimal perturbations than the single-step FGSM attack. Consequently, PGD robustness is widely regarded as the more challenging and informative benchmark.

It is important to note that LNL-MoEx evaluates robustness using perturbation strengths up to $\varepsilon = 1$, a very large budget that does not reflect realistic threat levels for safety-critical tasks such as traffic sign recognition. To enable a fairer comparison within their perturbation scale, we trained an FGSM-Neuro-Symbolic variant at $\varepsilon = 0.5$, which achieves 74.00% FGSM accuracy, closing the gap with LNL-MoEx-S (77.8%) despite our simpler architecture and minimal training overhead. However, when training a PGD-based Neuro-Symbolic variant at $\varepsilon = 0.5$, the optimization became unstable and failed to converge, indicating that very large-magnitude perturbations may require additional stabilization techniques. Although such a large perturbation budget is uncommon in standard threat models—and, as shown in Figure 3, large values of ε heavily corrupt the input image—we use $\varepsilon = 0.5$ solely for fair comparison with LNL-MoEx.

Compared to LNL-MoEx, our model offers several advantages:

Table 3. Comparison of adversarial accuracy under FGSM and PGD attacks across various transformer-based models with and without data augmentations, alongside our neuro-symbolic variants.

Method	FGSM Accuracy	PGD Accuracy
Without Augmentations		
PVT-Tiny[19]	10.8%	2.0%
TNT-T[20]	10.9%	3.9%
T2T-ViT-t-10[21]	10.2%	0.4%
RVT-Ti[22]	31.5%	13.9%
Swin-T[23]	20.2%	7.8%
PVT-Small[19]	21.4%	2.4%
TNT-S[20]	21.2%	6.8%
T2T-ViT-t-14[21]	21.7%	4.9%
RVT-S[22]	46.1%	25.3%
With Augmentations		
DeepAugment[24]	**65.1%**	42.2%
CutMix[25]	62.6%	40.1%
AugMix[26]	61.7%	39.8%
Puzzle-Mix[27]	**67.9%**	**44.2%**
RVT-Ti*[22]	38.9%	20.7%
RVT-S*[22]	51.2%	32.9%
Neuro-Symbolic (Ours)		
FGSM-Neuro-Symbolic ($\varepsilon = 0.03$)	63.10%	42.68%
FGSM-Neuro-Symbolic ($\varepsilon = 0.5$)	74.00%	25.00%
PGD-Neuro-Symbolic ($\varepsilon = 0.03$)	65.75%	56.00%

- **Interpretability**: While LNL-MoEx enhances robustness through architectural and statistical techniques, our neuro-symbolic model improves both robustness and interpretability by enforcing semantic rule consistency that provides transparent reasoning behind each prediction.
- **Lightweight Regularization**: Our symbolic supervision module is orthogonal to the model architecture and can be easily added to any neural backbone without architectural redesign or tuning of fusion strategies.
- **Efficient Training**: Our model achieves competitive adversarial performance using only 10 training epochs on GTSRB, while LNL-MoEx requires over 100 epochs and complex MoEx augmentation to reach similar results.

To further contextualize our results, Table 3 compares the adversarial performance of our neuro-symbolic variants against a wide range of transformer-based models with and without data augmentation, as reported in previous work[18]. In particular, many of these transformer baselines exhibit low robustness under FGSM and PGD attacks when trained without augmentations. Even with advanced augmentation strategies such as Puzzle-Mix and DeepAugment, our PGD-Neuro-Symbolic model achieves superior or comparable robustness (56.00% under PGD), while maintaining a lightweight architecture and minimal training cost. This highlights the efficacy of symbolic reasoning in enhancing robustness, without relying on large-scale transformer backbones or augmentation pipelines.

5.2 Inference-Time Computation Comparison

While previous sections focused on robustness and accuracy, we now compare the inference-time computational cost of our neuro-symbolic model against both adversarial training and transformer-based LNL-MoEx[18]. As mentioned before, our model uses a modified ResNet18 backbone followed by symbolic heads. This setup enables reasoning about both class predictions and symbolic consistency under perturbation and clean samples. Each symbolic head is a lightweight MLP, as previously described. The resulting inference-time computation cost is close to that of a standard ResNet18 with additional symbolic heads—much lower than transformer-based defenses. In contrast, adversarially trained models (e.g., PGD-trained ResNet18) require only a single forward pass and no symbolic reasoning, yielding slightly lower cost but no interpretability. Meanwhile, LNL-MoEx uses a ViT-based backbone and incorporates moment exchange logic and locality-aware constraints, leading to substantially higher inference-time complexity.

Despite relying on symbolic reasoning, our model remains significantly more efficient than transformer-based defenses. As shown in Table 4, our approach achieves a practical trade-off—maintaining strong robustness and logic-driven interpretability, with only a low increase in computation relative to standard ResNet18. As traffic sign recognition systems are integral

Table 4. Comparison of inference-time computational cost across architectures. Our model balances reasoning and robustness with moderate overhead.

Method	Backbone	Accuracy(PGD)	Inference Cost
Adversarial Training	ResNet18	38.65%	Very Low
LNL-MoEx[18]	ViT	59.4%	High
`NeuroShield`	ResNet18 + MLPs	56.0%	Low

components of autonomous driving pipelines, their deployment on devices with limited resources and constrained computational budgets is a critical consideration. To this end, reducing inference-time computation overhead is essential for practical, real-world adoption. One promising direction is to explore techniques such as *early exit mechanisms* or *model quantization*, as suggested in prior work on neural network optimization for edge deployment[28,29], to reduce computational complexity further while maintaining symbolic robustness and interpretability.

6 Conclusion

This paper introduced **`NeuroShield`**, a neuro-symbolic framework that enhances the adversarial robustness and interpretability of deep learning models by integrating symbolic rule supervision. Through extensive experiments on the GTSRB dataset, we demonstrated that our model outperforms both standard adversarial training and state-of-the-art transformer-based defense methods, while requiring fewer training epochs and no data augmentation. Our PGD-Neuro-Symbolic model achieved acceptable accuracy under FGSM and PGD attacks while maintaining its accuracy on the clean dataset. Additionally, our approach balances robustness and reasoning with moderate inference-time computation overhead, making it a promising candidate for real-world deployment. In future work, we aim to further adapt our framework for edge devices and energy-constrained environments, such as those used in autonomous vehicles, by exploring lightweight backbones, early exits, and quantized inference. Another direction is to extend symbolic reasoning beyond static rule checks toward chained logical inference, enabling the model to apply domain knowledge to dynamic driving scenarios and generate new symbolic rules from foundational ones. This would allow neuro-symbolic systems to assist in higher-level decision-making under uncertainty, moving toward generalizable and trustworthy AI in safety-critical applications.

Data Availability

The experiments in this study use the GTSRB dataset, which is publicly available through the following Kaggle repository. This dataset provides the raw training and testing images used in our experiments. All adversarial data (FGSM and PGD) used in this work were generated on the fly by the algorithms implemented in the accompanying code repository, and no additional datasets were created or hosted externally. The full code for generating this data and reproducing our experiments is available on the GitHub repository. Additional experimental artifacts (e.g., trained model checkpoints and logs) are available from the corresponding author upon reasonable request.

References

1. Wang, W., Yang, Y. & Wu, F. Towards data-and knowledge-driven ai: a survey on neuro-symbolic computing. *IEEE Transactions on Pattern Analysis Mach. Intell.* (2024).
2. Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **105**, 2295–2329 (2017).
3. Badreddine, S., Garcez, A. d., Serafini, L. & Spranger, M. Logic tensor networks. *Artif. Intell.* **303**, 103649 (2022).
4. Taha, I. A. & Ghosh, J. Symbolic interpretation of artificial neural networks. *IEEE Transactions on Knowl. Data Eng.* **11**, 448–463 (1999).
5. Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. & Mukhopadhyay, D. A survey on adversarial attacks and defences. *CAAI Transactions on Intell. Technol.* **6**, 25–45 (2021).
6. Khoshavi, N. *et al.* Shieldenn: Online accelerated framework for fault-tolerant deep neural network architectures. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 1–6 (IEEE, 2020).
7. Dong, H. *et al.* Neural logic machines. *arXiv preprint arXiv:1904.11694* (2019).
8. Shindo, H., Dhami, D. S. & Kersting, K. Neuro-symbolic forward reasoning. *arXiv preprint arXiv:2110.09383* (2021).

9. Xu, J., Zhang, Z., Friedman, T., Liang, Y. & Broeck, G. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, 5502–5511 (PMLR, 2018).

10. Xie, X., Kersting, K. & Neider, D. Neuro-symbolic verification of deep neural networks. *arXiv preprint arXiv:2203.00938* (2022).

11. Goodfellow, I. J., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

13. Athalye, A., Carlini, N. & Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, 274–283 (PMLR, 2018).

14. Carlini, N. & Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 3–14 (2017).

15. Cohen, J., Rosenfeld, E. & Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, 1310–1320 (PMLR, 2019).

16. Ahmed, K. *et al.* Pylon: A pytorch framework for learning with constraints. In *NeurIPS 2021 Competitions and Demonstrations Track*, 319–324 (PMLR, 2022).

17. Stallkamp, J., Schlipsing, M., Salmen, J. & Igel, C. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, 1453–1460 (IEEE, 2011).

18. Manzari, O. N., Kashiani, H., Dehkordi, H. A. & Shokouhi, S. B. Robust transformer with locality inductive bias and feature normalization. *Eng. Sci. Technol. an Int. J.* **38**, 101320 (2023).

19. Wang, W. *et al.* Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 568–578 (2021).

20. Han, K. *et al.* Transformer in transformer. *Adv. neural information processing systems* **34**, 15908–15919 (2021).

21. Yuan, L. *et al.* Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 558–567 (2021).

22. Mao, X. *et al.* Towards robust vision transformer. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 12042–12051 (2022).

23. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022 (2021).

24. Hendrycks, D. *et al.* The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8340–8349 (2021).

25. Yun, S. *et al.* Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032 (2019).

26. Hendrycks, D. *et al.* Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781* (2019).

27. Kim, J.-H., Choo, W. & Song, H. O. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International conference on machine learning*, 5275–5285 (PMLR, 2020).

28. Shafiee Sarvestani, A., Tabrizchi, S., Sehatbakhsh, N. & Roohi, A. Rl-sep: Rl-based s mart e xit point selection for enhancing energy harvested system longevity. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, 638–639 (2025).

29. Solanki, N., Tabrizchi, S., Sohrabi, S., Schmidt, J. & Roohi, A. Atm-net: Adaptive termination and multi-precision neural networks for energy-harvested edge intelligence. *arXiv preprint arXiv:2502.09822* (2025).

Funding

This work is supported in part by the National Science Foundation (NSF) under grant numbers 2504839, 2447566, and 2448133.

Author contributions statement

A. Sh. conceived the main idea and led the initial study design. A. Sh. and J. S. implemented the simulations, carried out the analyses, and prepared the initial manuscript draft. J. S. further refined and interpreted the results. A. R. contributed to developing the concept, critically reviewed the simulations and analyses, discussed and interpreted the findings, revised and proofread the manuscript, and supervised the overall study.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Arman Roohi.