Journal Title
XX(X):1–26

Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Neuro-LENS: a neuro-symbolic framework integrating incomplete background knowledge and deep learning

Giulia Murtas[1,2], Veselka Boeva[2] and Elena Tsiporkova[1]

Abstract

In this study, we propose Neuro-LENS, a Neuro-Symbolic Evidence-based Logic and Symbolic Reasoning framework that combines incomplete symbolic knowledge with neural learning to address ambiguity and improve the accuracy and interpretability of the results. We explore three strategies for integrating symbolic reasoning with deep learning and evaluate their effectiveness in practical settings: (i) applying the symbolic component to the neural output (neural-to-symbolic chaining); (ii) generating additional neural input features through symbolic rules (symbolic-to-neural chaining); (iii) creating an ensemble reasoning model (parallel neural-symbolic integration). The potential of the proposed Neuro-LENS framework is demonstrated on two real-world use cases: scene classification with abandoned object detection and prognostic health monitoring with vehicle failure prediction.

Keywords

Evidence measures, Modal logic, Multi-valued mapping, Deep learning

[1]EluciDATA Lab, Sirris, Ravensteinstraat 4, Brussels, Belgium
[2]Department of Computer Science, Blekinge Institute of Technology, Sweden

Corresponding author:
Giulia Murtas, Sirris, Brussels, Belgium and Blekinge Institute of Technology, Sweden.
Email: giulia.murtas@sirris.be, giulia.murtas@bth.se

1 Introduction

Deep learning has achieved remarkable results in perception-driven tasks such as image recognition, natural language processing, and fault detection in industrial systems. However, deep learning methods still suffer from lack of robustness and interpretability, and the difficulty of directly incorporating structured background knowledge [Mar18]. Symbolic logic, on the other hand, is apt for representing structured thought and explainable reasoning, although it struggles with scalability and perception tasks. This long-standing trade-off has motivated the development of neuro-symbolic integration, which aims to unify the learning capacity of neural networks with the structured reasoning power of symbolic systems [Bes+17].

Injecting reasoning abilities in artificial intelligence remains one of the central challenges in the field, as it would allow to enhance generalization and adaptability and produce explainable AI models which can perform logical inference, make decisions based on knowledge, and tackle structured problem solving [LWT25; BL04]. This hybrid approach has shown advantages over purely symbolic or purely neural systems, especially in real-world settings with noisy, unstructured data, as its flexibility makes it robust and well-suited for real-world AI applications [Bes+17].

Specifically in real-world industrial applications, neuro-symbolic approaches can help when dealing with noisy data and incomplete background knowledge. Traditionally, probabilistic models such as Bayesian networks and Markov decision processes are used to capture uncertainty and randomness in reasoning processes, while logic-based systems are exploited to model high-level reasoning and decision making. Evidence theory provides a bridge between the two paradigms, allowing to achieve high-level reasoning while dealing with uncertainty and incomplete knowledge, making its combination with deep learning approaches suited for real-world use cases.

In the current study, we propose a neuro-symbolic framework, Neuro-LENS (modal Logic and EvideNce-based Symbolic reasoning), based on evidence fusion, which integrates incomplete symbolic knowledge with neural learning in order to improve both accuracy and interpretability of the obtained results. Three strategies for integrating symbolic reasoning with deep learning in practical settings are explored:

(i) **Neural-to-symbolic chaining:** Applying symbolic reasoning on neural outputs to perform classification tasks;

(ii) **Symbolic-to-neural chaining:** Using symbolic reasoning to create new features that extend the neural input space, enabling more robust predictions and the integration of background knowledge/context;

(iii) **Parallel neural-symbolic integration:** Combining decision rules derived from both neural and symbolic components into a hybrid, rule-based classifier, providing improved interpretability.

This work builds upon and develops further the methodology presented in [MBT25]. In the paper, a novel neuro-symbolic approach was introduced, integrating modal logic, evidence theory, and deep learning, for the purpose of reasoning and decision making under ambiguity. The potential of the proposed hybrid method was validated on a real-world use case, more concretely, on scene classification for surveillance applications. In the current study, besides further enhancement and refinement of the theoretical framework, two new additional alternative mechanisms for the integration of modal logic, evidence theory, and deep learning are also considered.

The current work makes the following additional contribution with respect to the paper [MBT25]:

- The approach presented in the original paper is embedded within a framework integrating deep learning and symbolic reasoning.
- The theoretical background of the presented approach is extended.
- Two new strategies for the integration of a neural and a symbolic component are introduced.
- A completely new use case is studied for the validation of the two novel strategies.

Moreover, the current work aims to demonstrate that the applicability of the proposed neuro-symbolic approach is not limited to image data scenarios and can be generalized to completely different use cases dealing with data types of very different nature, e.g., specification records or time series sensor measurements.

2 Background

2.1 Multi-valued mapping

In this section, we introduce some basic concepts from the theory of multi-valued mappings [AF90; Ber77]. A *multi-valued mapping* $\mathcal{F}$ from a universe X into a universe Y associates to each element x of X a subset $\mathcal{F}(x)$ of Y. The *domain* of $\mathcal{F}$, denoted $\mathrm{dom}(\mathcal{F})$, is defined as

$$\mathrm{dom}(\mathcal{F}) = \{x \mid x \in X \ \wedge \ \mathcal{F}(x) \neq \emptyset\}.$$

$\mathcal{F}$ is called *non-void* if $(\forall x \in X)(\mathcal{F}(x) \neq \emptyset)$, i.e., if $\mathrm{dom}(\mathcal{F}) = X$.

Consider a subset A of X and a subset B of Y. The following direct and inverse images can be defined under multi-valued mapping $\mathcal{F}$:

(i) The *direct image* of A under $\mathcal{F}$ is the subset $\mathcal{F}(A)$ of Y, defined as

$$\mathcal{F}(A) = \bigcup_{x \in A} \mathcal{F}(x).$$

(ii) The *inverse image* of B under $\mathcal{F}$ is the subset $\mathcal{F}^{-}(B)$ of X, defined as

$$\mathcal{F}^{-}(B) = \{x \mid x \in X \ \wedge \ \mathcal{F}(x) \cap B \neq \emptyset\}. \tag{1}$$

(iii) The *superinverse image* of B under $\mathcal{F}$ is the subset $\mathcal{F}^{+}(B)$ of X, defined as

$$\mathcal{F}^{+}(B) = \{x \mid x \in \mathrm{dom}(\mathcal{F}) \ \wedge \ \mathcal{F}(x) \subseteq B\}. \tag{2}$$

(iv) The *subinverse image* of B under $\mathcal{F}$ is the subset $\mathcal{F}^{\sim}(B)$ of X, defined as

$$\mathcal{F}^{\sim}(B) = \{x \mid x \in X \ \wedge\ B \subseteq \mathcal{F}(x)\}.$$

(v) The *pure inverse image* of B under $\mathcal{F}$ is the subset $\mathcal{F}^{-1}(B)$ of X, defined as

$$\mathcal{F}^{-1}(B) = \{x \mid x \in X \ \wedge\ \mathcal{F}(x) = B\}.$$

A schematic visualization of the inverse and superinverse images, used in this work, is depicted in Fig. 1.

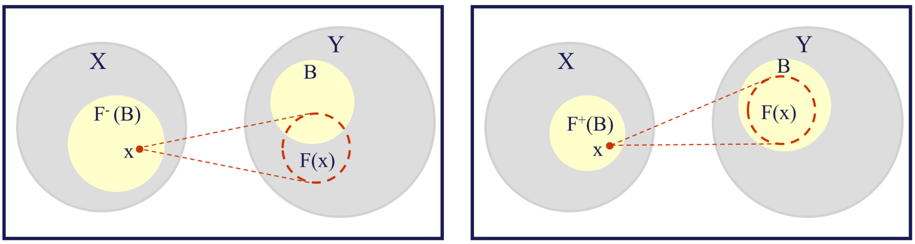

Figure 1. A visual illustration of inverse $F^{-}(B)$ (left) and superinverse $F^{+}(B)$ (right) images of a set B under a multi-valued mapping F from a set X into a set Y, which associates to each element x of X a subset $F(x)$ of Y. The figure is adapted from [MBT25].

2.2 Evidence measures

Evidence theory, also known as Dempster-Shafer theory, was initiated by Dempster with his study of upper and lower probabilities [Dem08]. He showed that if P is a probability measure on $\mathcal{P}(X)$, then a multi-valued mapping $\mathcal{F}$ from X into Y induces *upper* P^* and *lower* P_* *probabilities* on $\mathcal{P}(Y)$, as follows:

$$\begin{aligned} P^*(B) &= P(\mathcal{F}^{-}(B) \mid \mathrm{dom}(\mathcal{F})) \\ P_*(B) &= P(\mathcal{F}^{+}(B) \mid \mathrm{dom}(\mathcal{F})). \end{aligned} \tag{3}$$

It is clear that P^* and P_* are only well defined if $P(\mathrm{dom}(\mathcal{F})) > 0$. Note that P^* and P_* are dual, i.e., $P^*(B) = 1 - P_*(\mathrm{co}\, B)$.

Shafer reinterpreted upper and lower probabilities as degrees of *plausibility* Pl and *belief* Bel, abandoning Dempster's idea that they emerge as upper and lower bounds of Bayesian probabilities [Sha76]. Furthermore, in case of a finite universe Y, Shafer introduced the concepts of a basic probability assignment and its focal elements. Formally, a $\mathcal{P}(Y) \to [0,1]$ mapping m is called a *basic probability assignment* on $\mathcal{P}(Y)$ if $m(\emptyset) = 0$ and

$$\sum_{B \in \mathcal{P}(Y)} m(B) = 1.$$

A subset F of Y for which $m(F) > 0$ is called a *focal element* of m. The belief Bel and plausibility Pl measures can be defined in terms of basic probability assignment as follows:

$$\text{Bel}(B) = \sum_{C \subseteq B} m(C) \qquad \text{Pl}(B) = \sum_{C \cap B \neq \emptyset} m(C),$$

where, the corresponding basic probability assignment m is given by [Dem67]:

$$m(B) = P(\mathcal{F}^{-1}(B) \mid \text{dom}(\mathcal{F})). \tag{4}$$

2.3 Modal logic

Modal logic is an extension of classical propositional logic. It has been developed to formalize arguments that involve the notions of necessity and possibility [Che80]. These notions are often expressed using the concept of *possible worlds*: necessary propositions are those that are true in all possible worlds, whereas possible propositions are those that are true in at least one possible world. Possible worlds are abstract concepts, and it is difficult to provide a precise definition of them. Intuitively, however, we can view them as possible states of affairs, situations or scenarios.

The language of modal logic consists of a set of atomic propositions, logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, and modal operators of *possibility* $\Diamond$ and *necessity* $\Box$. The propositions of the language can be the atomic propositions, and if p and q are propositions, then are so $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$, $\Box p$, $\Diamond p$.

The interpretations of the Dempster-Shafer theory [TBD99; TBB00] used in this study are based on the semantics of modal logic using the concept of a standard model. A *standard model* of modal logic is a triplet $M = \langle W, R, V \rangle$, where W denotes a set of possible worlds, R is a binary relation on W called *accessibility relation*, and V is the *value assignment function* by which truth T or falsity F of each atomic proposition p in each world w is assigned. A proposition p may have different truth-values in different worlds. Therefore V assigns the truth-values not to proposition constants alone, but to pairs consisting of a possible world and a proposition constant, i.e., the value $V(w, p)$ is to be thought of as the truth-value of p in w. The value assignment function is inductively extended to all propositions in the usual way. The extension to possibilitations, i.e., propositions of the type $\Diamond p$, and necessitations, i.e., propositions of the type $\Box p$, are defined for any proposition p and any world $w \in W$ as follows:

$$V(w, \Diamond p) = T \Leftrightarrow \exists v \in W : wRv \wedge V(v, p) = T$$
$$V(w, \Box p) = T \Leftrightarrow \forall v \in W : wRv \Rightarrow V(v, p) = T.$$

2.4 Modal logic interpretations of evidence measures

Dempster-Shafer theory is closely related to the theory of multi-valued mappings as discussed above. In several studies [BTB98; TBB00; TBD99], set-valued interpretations of plausibility and belief measures in modal logic have been proposed. The authors consider a model $M = \langle W, R, V, P \rangle$, where P is a probability measure on the powerset

$\mathcal{P}(W)$ of W. Furthermore, the propositions have the form $e_A =$"*a given incompletely characterized element* ϵ *is classified in set* A", where $\epsilon \in X$ and $A \in \mathcal{P}(X)$. As atomic propositions, they consider the propositions $e_{\{x\}}$, for all $x \in X$. In addition, it is assumed that exactly one $e_{\{x\}}$ is true in each world. This implies that e_X and also $e_A \leftrightarrow \neg e_{\mathrm{co}\,A}$ are always true in M. In this context it is shown that a plausibility measure and a belief measure can be expressed in terms of conditional probabilities of truth sets of possibilities and necessities, i.e.

$$\mathrm{Pl}(A) = P(\|\Diamond e_A\|^M \mid \|\Diamond e_X\|^M)$$
$$\mathrm{Bel}(A) = P(\|\Box e_A\|^M \mid \|\Diamond e_X\|^M).$$

3 Related work

Neuro-symbolic approaches in the literature have been leveraged to obtain interpretable systems that are robust to uncertainty while maintaining accuracy. The integration of symbolic components alleviates the downsides of deep learning-based methods, improving their performance on reasoning tasks and providing them with explainability features.

Deep learning approaches dealing with image data have dominated the literature. However, recent advancements have demonstrated the potential of neuro-symbolic methods in various applications, even outperforming traditional neural models in tasks like question answering and image classification [Fit25]. Neuro-symbolic approaches have shown great value specifically in safety-critical fields such as surveillance, medical imaging, or autonomous systems, where it is paramount to employ trustworthy models. In [Lu+25], Logical Neural Networks (LNNs) are used to combine learnable parameters with logical operators. The networks incorporate first-order logic and are able to learn rule thresholds and weights from the training data. In [Wan+23], the challenge of lack of annotated image data is tackled. The work combines a pre-trained computer vision model, which extracts features from the unlabeled images, and an inductive logic learner module inferring logic-based rules that can be exploited for the annotation. A human in the loop is queried to confirm the labeling of uncertain samples and to improve the derived logic-based rules. The study delivers promising results, but the reached accuracy is not yet on par with the labeling of human experts, on which it still relies for feedback in the active learning portion of the method pipeline.

Evidence theory is often leveraged in the symbolic component of integrated hybrid systems to deal with uncertainty in the data. In [Zha+23], it is used to re-label the training set, by assigning ambiguous images to a meta-category, i.e., a subset of all possible categories, and selecting the meta-category with the highest degree of belief for each selected image. Ambiguous images are defined as samples showing features of multiple classes. The model is re-trained on the dataset updated with meta-categories, so that it can learn without overfitting to incorrect labels or misclassified examples.

The application of neuro-symbolic approaches to time series is also a challenging task that is being extensively researched. Time-series data are central to applications ranging from finance and healthcare to manufacturing, autonomous driving, and traffic management. In safety critical domains such as medicine and public security, interpretability in models is fundamental and only trustworthy approaches are likely to be adopted. Post-hoc methods such as SHAP ([LL17]) can provide an explanation of the model's output based on the input features that were most influential in a prediction, but do not really aid in understanding the underlying model mechanism.

Neuro-symbolic frameworks can reach intrinsic interpretability while balancing an accuracy trade-off in the final results. Neuro-symbolic rule-based approaches have been investigated in this regard. In [Wan+25], a model called TemporalRule is proposed to automatically learn Signal Temporal Logic rules for interpretable time series classification. The work aims at solving, by taking the temporal properties of the data into account, the discrepancy between discrete logical rules and continuous neural networks, which might make generated logical rules inconsistent with the decision process that needs to be carried out. The input time series is represented in three views: raw data, frequency-domain features, and derivative (rate of change between subsequent points), each capturing different temporal properties. After having been binarized, the inputs are passed to a Temporal Logical Layer, where temporal operators (Always, Eventually, Until, and their combinations) are simulated using small neural networks. A Logical Layer combines temporal predicates using logical connectives (AND/OR), and a final Linear Layer assigns weights to the learned rules and generates the classification output. So far, the method has only been tested on univariate time series.

Dhont et al. ([DMT25]), again put an emphasis on interpretability, employing a hybrid modeling framework for traffic dynamics forecast in terms of humanly interpretable traffic states. The work proposes three different workflows: a purely neural approach leveraging CNNs or RNNs, a neural-to-symbolic one where a deep learning model detects current traffic state probabilities, which are then fed into Markov chains for the forecast, and a symbolic-to-neural one, where the detected traffic state probabilities form the input for a deep neural predictor performing the forecast. The purely neural model achieved the highest accuracy; the neuro-symbolic models, while performing slightly worse in accuracy, provide interpretability, computational efficiency, and easier adaptability. In addition to [Wan+25], the workflows are applied to multi-variate time series. The sequential nature of the proposed neuro-symbolic approaches makes them subject to a possibly compounding error; in the symbolic-to-neural models in particular, the final performance is highly dependent on the quality of the initial state detection step.

Hogea et al. ([Hog+24]) integrated logical rules into recurrent neural networks to improve interpretability and accuracy in fault diagnosis of gearboxes. The authors introduced LogicLSTM, which adds an Explainability Layer and a Logic Tensor Network (LTN) on top of a pre-trained LSTM model. The Explainability Layer re-weights the features based on feature importance, forcing the model to focus more on signals which are relevant for the task; in the LTN, logical rules derived from domain knowledge are introduced, and the network is further trained to maximize both predictive accuracy and logical consistency with the provided constraints. The method is best suited for scenarios where there is prior knowledge about the relationship between classes or numerical values. LogicLSTM performed better than the presented purely neural baselines, confirming the effectiveness of the addition of symbolic constraints to enhance model robustness in noisy environments. However, the method's performance is critically impacted by the number of available samples within each considered sequence of data; moreover, manual intervention seems to be required to define the leveraged logical rules.

4 Method: Neuro-LENS Framework

In this section, we provide a detailed explanation of the two main components (symbolic and neural) of our neural-symbolic approach, Neuro-LENS. We also explain how these components can be integrated into a neuro-symbolic learning framework to tackle different use case scenarios.

Neuro-LENS, as typically neuro-symbolic systems are, is characterized by *modularity* and *hierarchical organization*. Modularity relates to the construction of a neuro-symbolic network as an ensemble of neural networks, leading to more flexibility, simplicity, and maintainability. Hierarchical organization means that each subsequent network level uses the output of the preceding level as input, thus increasing the abstraction level of the model [GLG09].

The symbolic module (component) exploited within the Neuro-LENS workflows is outlined in detail in Section 4.1. The neural component to be used needs to be selected based on the type of data to be processed, the scope, and requirements of the considered use case. Pre-trained, off-the-shelve models or customized models can be employed. Three possible hierarchical organizations (workflows) of the two modules are proposed in Section 4.2. The first strategy, neural-to-symbolic chaining, was first presented in [MBT25]. The two additional alternative strategies presented in the current study, symbolic-to-neural chaining and parallel neural-symbolic integration, are novel extensions of [MBT25]. A high-level schematics of the three approaches is presented in Fig. 2.

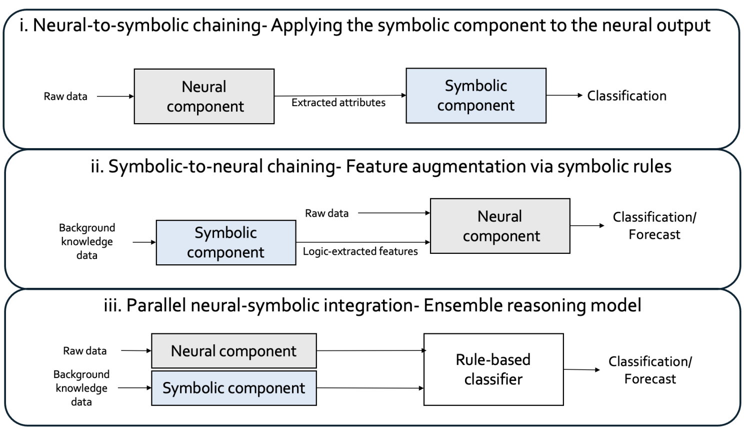

Figure 2. Three strategies for integrating neural and evidence-based (symbolic) components: (i) The neural component extracts attributes for use by the symbolic component; (ii) The symbolic component generates additional input features for the neural component; (iii) Both the symbolic and the neural components are used to extract inputs for a rule-based classifier.

4.1 Symbolic component

The symbolic component exploits modal logic and evidence theory in order to extract measures to quantify the uncertainty embedded in the raw data itself or in the available background knowledge. In brief, binary attributes of the considered samples are extracted to construct logical constraints that need to be satisfied by a sample to belong to a certain class, with a degree of uncertainty specified by its plausibility and belief measures. These measures can be used as such or combined into a single score, leveraged directly for interpretable classification, or fed to a neural network for further processing. A schematic view of the steps followed within this component can be consulted in Fig. 3.

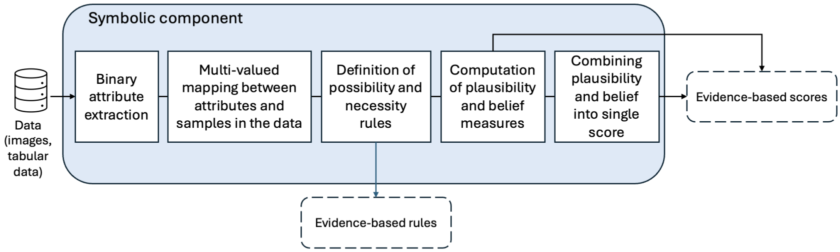

Figure 3. The symbolic component pipeline.

More concretely, multi-valued interpretations of upper and lower probabilities in modal logic are employed in order to reason within ambiguous scenarios. Consider a set of entities (objects) Y described by a set of attributes (properties) X. Each entity may have multiple properties, and a property may be associated with multiple entities. In addition, the entities in Y are distributed across c different categories (classes), i.e., $Y = \bigcup_{i=1}^{c} Y_i$, where $Y_i \subset Y$ and $Y_i \cap Y_j = \emptyset$, for $i \neq j$. In this scenario, our aim is to interpret each class in terms of its associated properties with the aim to enable automatic recognition of the most probable class of a new, unseen entity described by its properties.

In the above context, a multi-valued mapping $\mathcal{F}$ from the set of properties X to the set of entities Y can be defined. This mapping associates each property $x \in X$ with a set of entities $\mathcal{F}(x) \subseteq Y$ that possess it. The properties are defined as binary attributes that can be either satisfied or not by an entity. In the general case of multi-class classification, the mapping $\mathcal{F}$ is exploited to characterize each class Y_i, for $i = 1, 2, \ldots, c$, in terms of its possibility and necessity conditions, by constructing inverse and superinverse images of the class as defined in (1) and (2). Formally, the necessity and possibility conditions referring to class Y_i can be described by the following two expressions:

$$\Box Y_i = \bigvee_{x_j \in \mathcal{F}^{+}(Y_i)} x_j \quad \text{and} \quad \Diamond Y_i = \bigvee_{x_j \in \mathcal{F}^{-}(Y_i)} x_j. \tag{5}$$

Intuitively, a property x_j contributes to the possibility condition of a class Y_i, if at least one entity in its direct image $\mathcal{F}(x_j)$ satisfies this property of class Y_i. Similarly, a property x_j contributes to the necessity condition of class Y_i if all entities in its direct image under the function $\mathcal{F}$ satisfy this property of class Y_i. This reasoning is repeated for all defined properties and the final possibility and necessity conditions for class Y_i are defined as the disjunction of all single properties contributing to each of them.

The inferred possibility and necessity conditions of the classes defined in (5) can be used to reason about, and eventually predict, the most probable class of unseen entities, based on their properties. In addition, the plausibility and belief that each new unseen entity belongs to each class Y_i (for $i = 1, \ldots, c$) can be computed. The plausibility (Pl_i) and belief (Bel_i) that an entity presented by a set of properties X_j belongs to class Y_i are computed as the ratio of instances that satisfy the possibility and necessity conditions of the class, as follows:

$$\mathrm{Pl}_i(X_j) = |\ \Diamond Y_i(X_j)\ |\ /\ |\ \Diamond Y_i\ | \quad \text{and} \quad \mathrm{Bel}_i(X_j) = |\ \Box Y_i(X_j)\ |\ /\ |\ \Box Y_i\ |. \tag{6}$$

The calculated plausibility and belief values can be used to extend the feature set in the proposed integration strategy (ii) discussed in Section 4.2. These values can also be combined to calculate a single score for each entity-class pair. Namely, a scoring function S can be defined, which combines the plausibility and belief measures for all classes, producing a value in the interval $[0, 1]$ that can be interpreted as the likelihood of an entity X_j to belong to a certain class. More concretely, the calculated beliefs and plausibilities for a given entity represented by the set of properties X_j, with respect to a set of classes c, form a set of intervals, $[\mathrm{Bel}_k(X_j),\ \mathrm{Pl}_k(X_j)]$, for $k = 1, \ldots, c$. The width of these intervals is correlated with the uncertainty associated with that an entity presented by X_j belongs to class Y_i, respectively. Thus a scoring function S_i expressing this uncertainty can be defined as ratio of available evidence supporting class Y_i:

$$S_i(X_j) = \frac{\mathrm{Pl}_i(X_j) + \mathrm{Bel}_i(X_j)}{\sum_{k=1}^{c} \big(\mathrm{Pl}_k(X_j) + \mathrm{Bel}_k(X_j)\big)}. \tag{7}$$

Note that the above scoring function is a generalization to multi-class context of the scoring function defined in [MBT25] in case of a binary classification task, i.e., two classes, positive $(+)$ and negative $(-)$:

$$S(X_j) = \frac{\mathrm{Pl}_+(X_j) + \mathrm{Bel}_+(X_j)}{\big(\mathrm{Pl}_+(X_j) + \mathrm{Bel}_+(X_j)\big) + \big(\mathrm{Pl}_-(X_j) + \mathrm{Bel}_-(X_j)\big)}. \tag{8}$$

4.2 Neuro-LENS: Neuro-symbolic integration

In the current section, we discuss the different ways in which background knowledge can be expressed in an evidence-based language and integrated into a neural / deep learning (DL) component to improve model performance. Three paradigms are presented within the Neuro-LENS framework:

(i) *Neural-to-symbolic chaining*: the symbolic component is applied to the neural output. This approach is relevant when the type of data cannot be processed by the symbolic component as such (e.g., images), and needs to first be transformed into a suitable representation. Thus, a DL model capable of processing the considered data type is applied to the raw data in order to extract features to be fed as input to the symbolic component described in Section 4.1. The latter exploits the extracted features to produce both logical rules and a score, used to perform respectively rule- or score-based classification, as demonstrated in [MBT25].

(ii) *Symbolic-to-neural chaining*: the symbolic component generates extra input features for the neural component. This strategy is valuable when the use case requires integrating data of different type, e.g., time series sensor measurements and background knowledge as configuration specifications or log events. The usage of background knowledge in industrial settings is often compromised due to the ambiguity typically present in such datasets, making them not immediately suitable for model learning. The developed symbolic module is capable of deriving relevant features from background knowledge datasets and of dealing with their ambiguity. The features are then used to enhance the feature input of a suitable DL model. The potential of this integration scheme is supported by our validation study on the Scania use case.

(iii) *Parallel neural-symbolic parallel integration*: symbolic and neural components create an ensemble reasoning model. Here, both components extract features from the data in parallel, allowing to employ the most suitable modeling paradigm for each data type and still be able to benefit from the interpretability of the symbolic component. The features are subsequently used to construct decision rules.

Finally, note that, like most data-driven methods, the Neuro-LENS method is sensitive to the variability and size of the training dataset. However, an important advantage of our method is that it can deal with imbalanced datasets. This allows an initial model to be bootstrapped from a limited set of available labels, which can easily be upgraded when more data becomes available.

5 Experiments and evaluation

In our experiments, we have used real-world datasets to simulate two use case scenarios: scene classification with abandoned object detection, and prognostic health monitoring with vehicle failure prediction. The following subsections present the two use cases, elaborating on the problem statement, data, method adaptation, and results of each of the explored domains.

5.1 Scene classification

Within this use case, we aim at applying the proposed neuro-symbolic framework to perform a binary scene classification task: understanding whether a frame taken from surveillance videos of train station or similar public places, contains (positive class) or does not contain (negative class) an abandoned luggage. The task usually involves complex scenarios and ambiguity, and it can be safety critical.

5.1.1 Datasets

The proposed method is validated on the PETS2006 and the AVS2007 datasets, both containing videos depicting abandoned luggage scenarios.

The PETS2006 dataset contains videos with multi-sensor sequences depicting scenes of a luggage being abandoned inside a train station. Static frames are extracted from the videos in order to apply the proposed approach. Ground truth is not available, neither for the object detection task or the abandoned bag scene classification task. Labels indicating whether the represented scene contains an abandoned bag have been manually identified and created. The dataset consists of 1325 images, of which 95% do not depict an abandoned object, while in the remaining 5% an abandoned bag can be detected.

The AVS2007 dataset (Advanced Video and Signal Based Surveillance) provides benchmark datasets for testing and evaluating detection and tracking algorithms. The i-LIDS bag subset of AVS2007 is considered, as it consists of abandoned luggage scenarios. The dataset comprises of 161 images, 14% of which shows an abandoned object. Again, labels indicating whether an abandoned bag is present in the image have been manually added to the data.

5.1.2 Method adaptation

The current section details the adaptation of the neural-to-symbolic chaining integration strategy to the scene classification use case. The process consists of three stages, as depicted in Fig. 6:

1. The available set of labeled images is fed to the neural component, the deep-learning based block of the framework. The neural component makes use of two pre-trained DL models: a OneFormer model [Jai+23], which detects the objects of interest in the images (in this case, people and bags) and returns the classes and bounding boxes of the detected objects; a Depth Anything V2 model [Yan+24], which provides a pixel-wise measure of the depth of each object in an image, allowing to more accurately place objects in a 2D image. Fig 4 shows the output of the two DL models on an example image.

Figure 4. Example of object detection (left) and depth estimation (right) results [MBT25]

2. The outputs of the deep learning models are passed to the symbolic component, which uses them to derive meaningful attributes (or instances) for the use case at hand, which characterize the input images. The attributes contain relevant information about the people and luggage depicted in the image and the relationship between them (e.g., overlap between their bounding boxes and

distance between a luggage and the person closest to it, Fig. 5. Then, a multi-valued mapping between the set of attributes and the images to be categorized is constructed, by associating each instance with the set of images in which it appears. A list of the extracted attributes can be seen in the first column of Table 1. Note that all extracted attributes are binary.

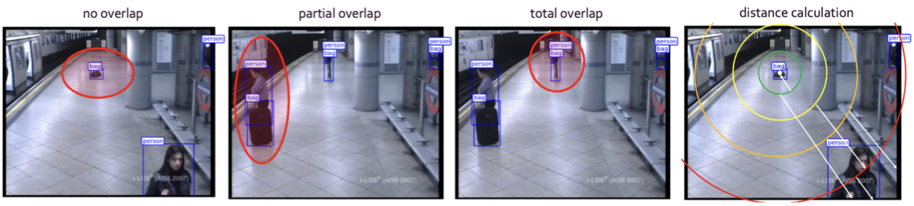

Figure 5. Overlap types and distance calculation for a selected bag [MBT25]. The distance between two objects is estimated as the distance between the centers of their bounding boxes while considering the estimated depth of each object, i.e., a 3-dimensional Euclidean distance is calculated. The calculated distances are binned into five overlapping ranges formed by increasing the radius of concentric circles with the bag of interest in their center.

3. The inverse and superinverse images of the multi-valued mapping are used to define the necessity and possibility conditions for the positive and negative classes. To exemplify, the necessity conditions for the positive class describe the attributes an image has when it depicts a scene *necessarily* containing an abandoned luggage. The possibility conditions for the positive class specify the attributes an image has if it depicts a scene *possibly* containing an abandoned luggage. The obtained conditions are presented in Table 1. It can be observed that eight instances contribute to the discrimination between the two classes in the PETS2006 dataset (one fewer in the AVS2007 dataset). The ambiguity aspect is captured by the instances which are common to the two classes, indicated below by their index: $ambiguous\_evidence = \Diamond\{\textit{abandoned}\} \cap \Diamond\{\textit{non-abandoned}\} = \{0, 1, 5, 7, 8\}$. Thus, the possibility of either of the two classes can be expressed, as shown below, as the disjunction of the respective necessity of this class and the ambiguous evidence:

$$\Diamond\{\textit{abandoned}\} = \Box\{\textit{abandoned}\} \vee ambiguous\_evidence$$

$$\Diamond\{\textit{non-abandoned}\} = \Box\{\textit{non-abandoned}\} \vee ambiguous\_evidence.$$

Next, the decision rules exploited by the rule-based classifier are defined. An image is assigned to the positive (negative) class if the instances representing it satisfy the necessity conditions for the positive (negative) class, i.e.,

$$\begin{array}{l} \text{IF } \Box X_+(X_i) \text{ THEN } X_i \in \text{positive class} \\ \text{IF } \Box X_-(X_i) \text{ THEN } X_i \in \text{negative class,} \end{array}$$

Table 1. Inverse ($poss_+$, $poss_-$) and superinverse (nec_+, nec_-) images for the two classes.

attributes	$poss_+$	$poss_-$	nec_+	nec_-
0: contains_bag	**True**	**True**	False	False
1: contains_person	**True**	**True**	False	False
2: contains_person_but_no_bag	False	**True**	False	**True**
3: has_partial_overlap	**True*** | False	**True**	False	False* | **True**
4: has_total_overlap	False	**True**	False	**True**
5: has_no_overlap	**True**	**True**	False	False
6: min_distance_below_0.1	False	**True**	False	**True**
7: min_distance_above_0.1	**True**	**True**	False	False
8: min_distance_above_0.25	**True**	**True**	False	False
9: min_distance_above_0.5	**True**	False	**True**	False
10: min_distance_above_0.75	**True**	False	**True**	False

* These are the values for the AVS2007 data set. All other values are the same for both datasets.

where $\Box X_+$ and $\Box X_-$ are the necessity conditions of the positive and negative classes, respectively. Consequently, in the context of our use case, Table 1 can be used to define the decision rules for the two classes as follows:

$$\begin{array}{l}\text{IF } (9 \vee 10) \text{ THEN } x \in \{abandoned\} \\ \text{IF } (2 \vee 3 \vee 4 \vee 6) \text{ THEN } x \in \{non\text{-}abandoned\}.\end{array} \tag{9}$$

In case an image does not satisfy either decision rule, it is assigned to a "none of known" class, in order to avoid misclassifications.
The necessity and possibility are further exploited to compute the plausibility and belief values for the two classes, using (6). These values are then combined into a single score using (8). The computed scores focus on the positive class, indicating the likelihood of an image to contain an abandoned luggage.

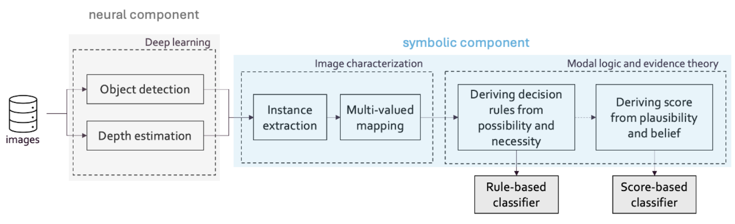

Figure 6. A schematic illustration of the first integration strategy as applied to scene classification task.

5.1.3 Results and discussion

The neural-to-symbolic strategy is applied to the two image datasets: AVS2007 and PETS2006 (see Section 5.1.1). The datasets are split into training and test sets with proportion 80/20, with the ratio of images belonging to each class kept fixed in the division. The extracted decision rules in (9) are employed to construct a rule-based classifier which assigns every image to its class according to the decision rule it satisfies. A "none of known" class is also created for ambiguous images which do not satisfy either decision rule. Table 2 reports the averaged results of the classifier over 20 iterations. Note that no sample is misclassified. Thus, we report as metric the percentage of samples which are considered ambiguous by the model.

Table 2. Performance of the rule-based classifier on the two datasets.

Metric (%)	AVS2007	PETS2006
Detected positives	50	76.7
Detected negatives	60.2	97.7
Positives in "none of known"	50	23.3
Negatives in "none of known"	39.8	2.3
Overall in "none of known"	41	3.5

A single score is obtained by combining plausibility and belief using (8), that quantifies the likelihood of an image to depict abandoned luggage. In Fig. 7 the ROC curves produced by the score-based classifier on the two datasets are depicted, illustrating the variation in the model's performance when varying the single score threshold. As can be seen, the classifier demonstrates good performance on both PETS2006 and AVS2007, showing to be robust to data scarcity and complexity of scenarios. The AVS2007 dataset is, in fact, much smaller than the PETS2006 dataset and contains more complex scenarios.

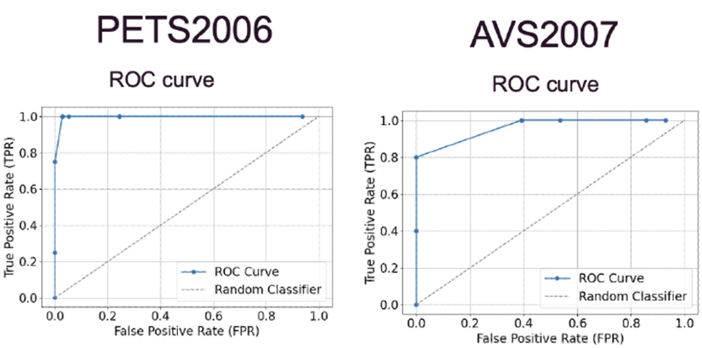

Figure 7. ROC curves of the score-based classifier for the two datasets.

Both rule-based and score-based classifiers demonstrate a strong discrimination potential in distinguishing between the two classes. The rule-based classifier allows to avoid misclassifications and signals uncertain scenarios which cannot be assigned to any class due to lack of sufficient evidence. The score-based classifier quantifies the risk of an abandoned object being present in a scene in a more granular fashion, offering the possibility of setting a threshold to trigger alarms for high-risk scenarios.

In summary, the neural-to-symbolic chaining strategy allows to leverage pre-trained DL models to extract attributes from images, which are subsequently used to obtain robust logical rules through the usage of modal logic and evidence theory. The resulting approach is significantly less sensitive to data scarcity and imbalance than fully DL-based methods, confirming the added value of neural and symbolic integration. Note also that the performance of the object detection DL model poses an upper bound to the performance of the overall framework. Thus, it is important to ensure that the object detection models deliver satisfactory results.

Baseline comparison

Considering the nature of the use case, it is difficult to find a suitable baseline study in the literature. Therefore, we built our own baseline rule-based classifier which does not leverage modal logic and the extracted instances and simply sets a threshold on the minimum distance between the considered bag and any person in an image. The threshold is selected based on the training set, and chosen to be the minimum distance observed in the set between an abandoned bag and a person present in the frame. We compare the complete framework to the simple rule-based model, to showcase the value of the proposed symbolic component. In Table 3, the precision, i.e., the accuracy of the positive predictions made by the two models, is shown for the AVS2007 and the PETS2006 datasets.

Dataset	Model	Precision (%)
AVS2007	Simple rule-based	0.25
	Neuro-LENS	1
PETS2006	Simple rule-based	0.52
	Neuro-LENS	1

Table 3. Precision of a basic rule-based model compared to Neuro-LENS

Moreover, we investigated the contribution of considering the estimated depth when calculating the distance between objects. The model's performance did not change for the AVS2007 dataset when the estimated object depth was not considered, while a clear performance degradation was observed when applying the model to the PETS2006 dataset without considering depth.

5.2 Vehicle failure prediction

This use case is concerned with the prediction of failures in a large fleet of over 23.000 heavy-duty trucks. The application context is completely different than the scene classification use case. Moreover, the described application deals with multi-source data (sensor measurements and technical specifications) and tackles a complex industrial phenomenon.

5.2.1 Dataset

Within the vehicle failure prediction use case, the **symbolic-to-neural chaining** and the **parallel neural-symbolic integration** strategies are applied and validated on the Scania dataset [Kha+25]. The latter is a real-world multi-source dataset collected from a single engine component across a fleet of SCANIA trucks. The dataset contains: operational data collected by onboard sensors; repair records, which include information about maintenance, repairs, and servicing performed on the vehicles; specifications of the analyzed component, collected with the production system, such as engine type, weight capacities, dimensions, and other technical details. The operational data are stored as multi-variate time series where the time steps are chronologically sequential but do not have a specified duration, and the amount of time they encompass can vary from one truck to another. As mentioned above, the specifications describe the technical properties of each truck. However, as the dataset is fully anonymized, the specifications are not indicated by name but merely numbered. Most trucks do not experience a fault during the observation period covered by the dataset. Table 4 shows the percentage of trucks that did or did not require maintenance in the training, validation, and test set. The high imbalance of healthy vs. faulty behavior constitutes a challenge for training a robust fault prediction model on this dataset. Each truck is annotated with a class label. The provided

Dataset	Healthy trucks (%)	Faulty trucks (%)
Train set	90.4	9.6
Validation set	97.3	2.7
Test set	97.2	2.8

Table 4. Percentage of healthy and faulty trucks.

class labels divide the dataset into 5 groups, based on the remaining time before failure as specified in Table 5. The last column of Table 5 contains the class label distribution in the training set, which demonstrates very clearly the highly imbalanced label context of this use case.

Class labels	Time to failure	Training set distribution (%)
Class 0	more than 48 hours left	90.4
Class 1	between 48 and 24 hours	0.1
Class 2	between 24 and 12 hours	0.3
Class 3	between 12 and 6 hours	0.7
Class 4	less than 6 hours left	8.5

Table 5. Truck class labels: meaning and distribution in the training set.

5.2.2 Method adaptation

In this section, we present how the symbolic-to-neural chaining and the parallel neural-symbolic integration strategies are applied in the context of the vehicle failure prediction use case. As the symbolic component is the same in the two strategies, it is only presented

Table 6. Inverse ($poss_+$, $poss_-$) and superinverse (nec_+, nec_-) images for the two classes.

attributes	$poss_+$	$poss_-$	nec_+	nec_-
$Spec_0 = Cat_0$	**True**	**True**	False	False
...	...	...	...	...
$Spec_1 = Cat_{14}$	False	**True**	False	**True**
...	...	...	...	...
$Spec_0 = Cat_0$ & $Spec_1 = Cat_0$	**True**	**True**	False	False
...	...	...	...	...
$Spec_0 = Cat_0$ & $Spec_1 = Cat_{16}$	**True**	False	**True**	False
...	...	...	...	...
$Spec_0 = Cat_0$ & $Spec_1 = Cat_0$ & $Spec_2 = Cat_0$	**True**	**True**	False	False
...	...	...	...	...
$Spec_0 = Cat_0$ & $Spec_1 = Cat_0$ & $Spec_2 = Cat_4$	False	**True**	False	**True**
...	...	...	...	...

once, in the paragraph below. The rest of the section describes the adaptation of the two strategies to the use case at hand.

Symbolic component

The symbolic component is used to derive the predisposition to failure of a vehicle, solely based on its technical characteristics. In the data, each vehicle is described by 8 specifications. The possible values each specification can take are expressed as numbered categories. Labels are available for each vehicle, indicating whether it has experienced a failure during the observation period of the dataset. Thus, as in the previous use case, the symbolic component deals with two classes: healthy (negative class, as in ”vehicles not presenting failures”) and failing vehicles (positive class). The steps carried out within the symbolic component are those seen in Fig. 3, similarly to the neural-to-symbolic strategy. The binary attributes needed for the approach (first block of the schema in Fig.3) are extracted by listing all possible combinations of specifications a vehicle can have. In order to take the interactions between different specifications into account, all possible pairs and triplets of specifications are also included in the attributes. A few examples of these attributes can be seen in the first column of 6. The process can theoretically be extended to larger groups of specifications, but the computational time quickly explodes.

Subsequently (second block in Fig 3, a multi-valued mapping is constructed between the set of extracted attributes and the set of vehicles in the training dataset. Namely, each specification, or combination of specifications, is mapped to all the vehicles in the training set that presents it. As in the first use case, the inverse and superinverse images of the constructed mapping characterize the two defined classes, and allow us to retrieve the possibility conditions (inverse image of the mapping function) and necessity conditions (superinverse image of the mapping function) of healthy and failing vehicles. A short excerpt of the obtained conditions for the two classes is shown in Table 6 for the purpose of illustration.

Using the formulas in (6), the plausibility and belief measures of the two classes can be computed from the obtained possibility and necessity conditions. As seen in Fig 3, these measures can directly be taken as output of the symbolic component and exploited in the following steps of the pipeline. Alternatively, they can be combined into a single score, using (8).

Symbolic-to-neural chaining
This section presents the second integration strategy in Fig. 2. Within this strategy, the outputs of the symbolic component are used as additional features for the neural component. The goal of adding the extra features is to infuse the neural component with the background knowledge contained in the data, which cannot otherwise be directly fed to the neural network. In the use case at hand, the provided background knowledge is the predisposition to failure of the trucks, based on their technical characteristics. The neural component uses an LSTM neural network, suited to process the multi-variate time series in the sensor data together with the output of the symbolic component to accurately predict failures in the vehicles.

Parallel neural-symbolic integration
In the third explored strategy, both neural and symbolic components are used to generate features, which are then combined into a rule-based model. The symbolic component remains the same as in the previous strategy. Within the neural component, an LSTM-autoencoder is trained to reconstruct sequences of time series data for healthy vehicles. The obtained reconstruction error together with the evidence metrics returned by the symbolic component are used to define the following logic rule, employed as a classifier

$$\text{IF } reconstruction\_error > x \; and \; (Pl_+ > y \; or \; Bel_+ > 0) \text{ THEN } failure\_detected,$$

where x and y are chosen based on the validation data distribution.

5.2.3 Results and discussion
In this section, the results obtained by applying the described frameworks on the SCANIA dataset are presented and discussed in detail.

Symbolic component
The symbolic component extracts the possibility and necessity conditions for the two classes based on the data in the original training and validation sets of the SCANIA dataset. The computed conditions are then used to calculate the plausibility and belief of the two classes for each vehicle in the test set. The ROC curves in Fig. 8 visualize the performance of classifying healthy and failing trucks when only exploiting the information contained in the specification data (technical characteristics of the trucks) via the extracted single score from (8), without taking the sensor data into account. The performance is compared to the risk score from [FTB24]. Here, the authors carry out a survival risk analysis on the vehicles in the SCANIA dataset. As part of their pipeline, they compute a risk score based on specification data using Cox Proportional Hazard analysis and survival trees. As the authors do not compute the bias on the test data, this comparison was only possible on the training set.

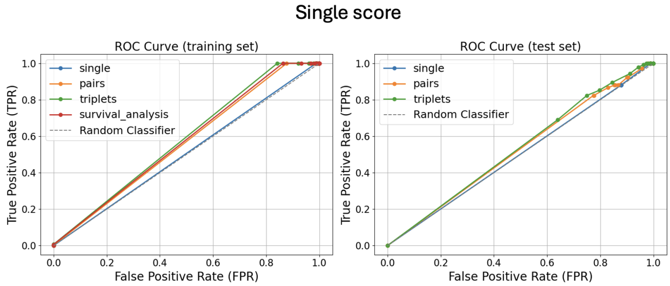

Figure 8. Classification performance on the training and test set when only using the symbolic component. The different traces show the results obtained when: 1)including only single specifications in the extracted attributes, 2) including single and pairs of specifications, 3) including single, pairs, and triplets of specifications. The 4th trace shows the performance obtained with the risk score from [FTB24]. The performance of a Random Classifier is also shown as reference.

Within the single score, we analyzed the weight of belief and plausibility values, by looking at the classification performance obtained when removing one or the other value from the score. Belief measures do not seem to contribute much in distinguishing between the classes. This is probably due to the fact that the technical characteristic of a truck only contain limited information about an eventual failure. Thus, it is highly unlikely that one (or a combination of) specific technical characteristic alone will necessarily (with high certainty) lead to failure. In other words, decisions taken when considering the specification data by itself are too uncertain in order to be able to benefit significantly by the belief measures.

Symbolic-to-neural chaining

As mentioned above, the **neural component** used within this strategy exploits an LSTM model, suited for handling sequential data. The model takes windows of 12 time steps as input and predicts one of the five class labels in presented in Table 5 for each of the 12 future time steps. The missing data in the training set are handled by performing forward filling, and the training and test set are defined in the same way as for the symbolic component. In order to validate the contribution of the symbolic component's output, the model is first trained on the sensor data only. In Fig. 9, this model is indicated with the label "Sensor data". The experiment is then repeated by adding the features produced by the symbolic component: either the plausibility and belief measures of the positive class (failing vehicles), or the combined single score. The addition of the symbolic component features is done in two alternative ways:

1. The features are simply concatenated to the input of the model, i.e., the same value is repeated at each time step, as the outputs of the symbolic model represent a static

property of the vehicles. The models where this approach was used are indicated with ”concatenated” in Fig. 9;

2. The features are used to set the LSTM initial state, allowing to treat them as a context rather than features replicated across all time steps. Additionally, a gate is added to the static features in order to avoid their influence being too strong compared to the time-series signal. In this manner, the network can learn how much weight to assign to the static features. The models indicated with ”gated” in Fig. 9 have been trained this way.

All evaluated models exhibit a very high accuracy (over 95is highly imbalanced, a high accuracy is not very indicative. Predicting imminent failures correctly (classes 3 and 4) is more important for the use case at hand than correctly individuating normal conditions (class 0). Thus, F1- score and precision are selected to compare the performance of the models. In Fig. 9, the evolution of the two metrics across all time steps is depicted, for class 0 (normal operation) and class 4 (imminent failure). The difference in performance between the two classes in the figure is striking, highlighting once again the challenge of dealing with an imbalanced dataset. When predicting normal operation (class 0, the majority class), all models perform well, and augmenting the LSTM’s input with the symbolic output does not provide measurable improvement. The overall performance for class 4 reflects the difficulty of predicting failures in the vehicles.

The model trained with the addition of the single score (gated) slightly outperforms the rest of the models, which might indicate that the plausibility and belief measures of the negative class, included in the formula for the computation of the single score (see (8)), also provide some relevant information. In addition, the fact that feeding the single score as a separate gated input to the network performs better than simply concatenating the score to the sensor data, supports the understanding that a more context-aware integration yields better results than naive concatenation.

Parallel neural-symbolic integration

In this integration strategy, the neural component includes an LSTM-autoencoder, trained on entire data sequences from healthy vehicles, i.e., vehicles that do not experience a failure during the monitored time period. 80% of healthy vehicles data from the SCANIA training set are used to train the autoencoder. The remaining 20% of healthy trucks is combined with the data from failing trucks which to form the validation set. The autoencoder is trained to reconstruct sequences of sensor data and takes windows of 48 time steps as input. Fig. 10 depicts the evolution of the average reconstruction error of healthy vs. failing vehicles on the validation set defined above, as we get closer to either a failure or the end of the observation period. The average reconstruction errors for failing trucks starts deviating already about 60 timesteps before a failure or end of observation. It is interesting to notice how the reconstruction error of healthy trucks also tends to increase in time. This confirms the inconsistency between the available discrete class labels and the actual continuous health degradation of a vehicle throughout its life. It must also be noted that such a sharp separation between failing and healthy vehicles

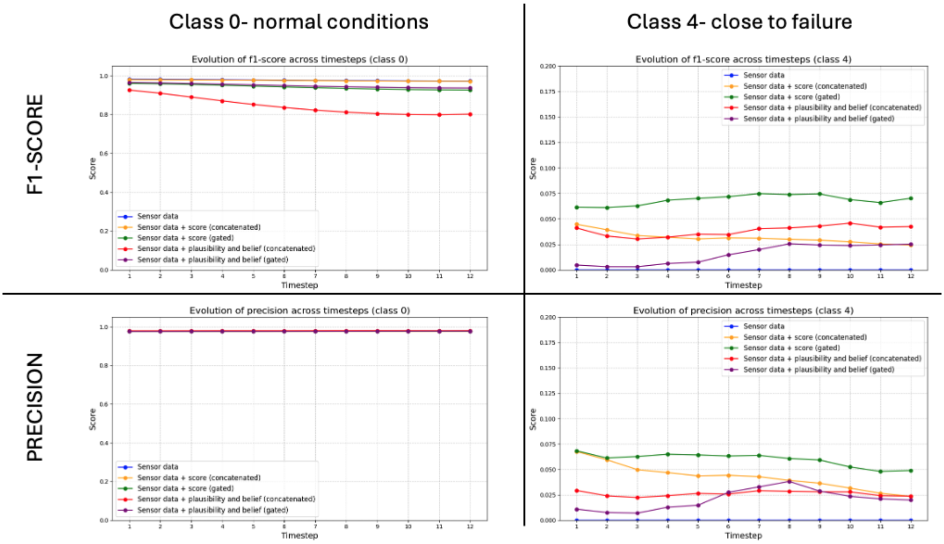

Figure 9. Class 0 vs. class 4 performance comparison in terms of F1-score and precision.

can only be detected in the average values, while there is a significant overlap between the two when looking at the values for single trucks.

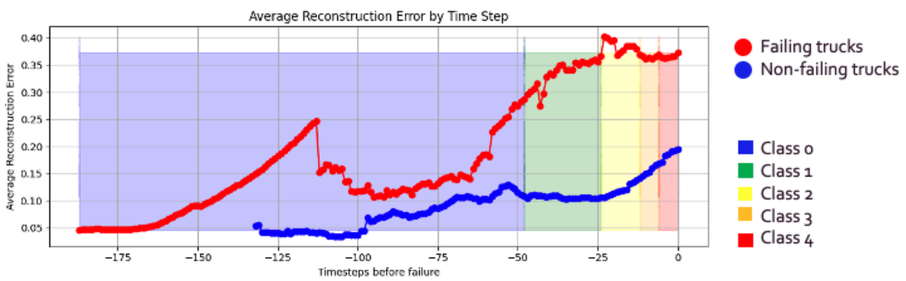

Figure 10. Evolution of the LSTM-autoencoder average reconstruction error for failing and non-failing vehicles as either a failure or the end of the observation window get closer. The time windows corresponding to the class labels defined in 5 are also shown.

The reconstruction error obtained by the autoencoder is combined with the outputs of the symbolic component and exploited by the rule-based classifier described in the previous section. Fig 11 depicts the performance of the rule-based classifier, confirming that very high accuracy can be obtained for vehicles operating in normal conditions, i.e., class 0, since the dataset is heavily skewed toward such samples. In order to optimize the number of predicted failures, a reconstruction error threshold of 0.2 is selected (prioritizing a higher recall even if losing accuracy) corresponding to an overall accuracy of over 80%.

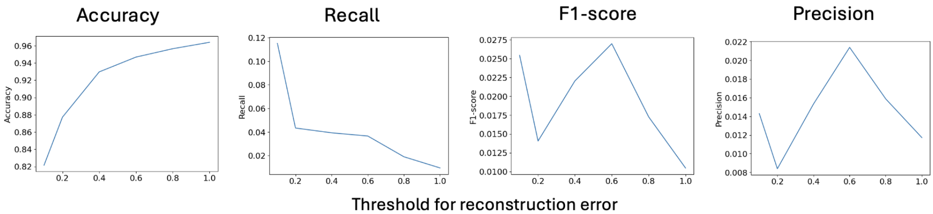

Figure 11. Rule-based classifier performance as function of the reconstruction error.

Fig. 12 shows the prediction performance of the rule-based classifier using the formula provided in the previous section, with the selected threshold for the reconstruction error, for different time horizons. As expected, the performance of the model decreases for longer time horizons.

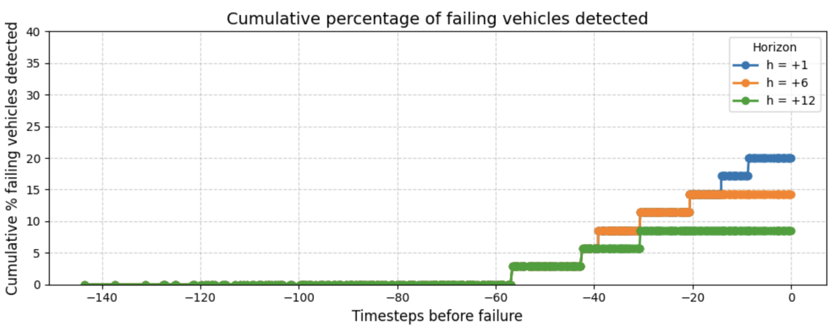

Figure 12. Cumulative percentage of failing vehicles correctly detected. Three temporal horizons are considered: prediction made 1 time step in advance, 6 time steps in advance, or 12 time steps in advance.

Baseline comparison

In this section, we compare the parallel neural-symbolic variation of the proposed approach with several alternative approaches presented in [ZW24], where the authors apply a variety of deep learning models to the SCANIA dataset to predict faulty trucks. These models were only trained on the time series data, discarding the specification data of the trucks. For this comparison, our approach was trained and evaluated on the same dataset split as indicated in the original paper [ZW24]. Table 7 presents the accuracy of the compared models. It is impressive to observe that exploiting the integration of the neural and symbolic components enables a simple LSTM to outperform the more complex deep learning architectures used in [ZW24]. This demonstrates the prediction potential of the Neuro-LENS framework and the importance of effectively embedding background knowledge in the model.

Model	Accuracy (%)
CNN	88.2
Bi-LSTM	77.6
Bi-LSTM with attention	81
Neuro-LENS	**93.3**

Table 7. The accuracy of the best-performing models presented in [ZW24] is benchmarked against the parallel neural-symbolic version of Neuro-LENS, as evaluated on the SCANIA validation set.

6 Conclusion

This study presented Neuro-LENS, a neuro-symbolic framework exploring different integrations of a neural model with a symbolic reasoning component. The symbolic component is based on the multi-valued interpretations of the theory of evidence in modal logic. As demonstrated in our experiments, the modular nature of the Neuro-LENS framework allows for great flexibility and generalizability across various use cases and data types. Three complementary strategies that integrate symbolic reasoning and deep learning have been investigated in our study: neural-to-symbolic chaining, symbolic-to-neural chaining, and parallel neural-symbolic integration. The Neuro-LENS framework has been validated across two different use cases (scene classification and failure prediction) using different data types (images vs. time series and tabular data). The results obtained highlight the framework's generalizability and practical relevance in real-world scenarios and industrial settings, where explainability, robustness, and trust are fundamental.

7 Acknowledgments

This research was partially funded by the Flemish Government through the AI Research Program.

Veselka Boeva's research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

References

[Ber77] Claude Berge. *Topological spaces: Including a treatment of multi-valued functions, vector spaces and convexity*. Oliver & Boyd, 1877.

[Dem67] A. Dempster. "Upper and lower probabilities induced by a multivalued mapping". In: *The Annals of Mathematical Statistics* 38 (1967), pp. 325–339.

[Sha76] G. Shafer. “A Mathematical Theory of Evidence”. In: Princeton University Press, Princeton, 1976.

[Che80] B. Chellas. “Modal Logic, an Introduction”. In: Cambridge University Press, Cambridge, 1980.

[AF90] J.-P. Aubin and H. Frankowska. “Set-Valued Analysis”. In: Birkhäuser, Boston–Basel–Berlin, 1990.

[BTB98] V. Boeva, E. Tsiporkova, and B. De Baets. “Modelling uncertainty with kripke’s semantics”. In: *Artificial Intelligence: Methodology, Systems, and Applications. AIMSA 1998. LNCS*. Ed. by F. Giunchiglia. Vol. 1480. Springer, Berlin, Heidelberg, 1998.

[TBD99] Elena Tsiporkova, Veselka Boeva, and Bernard De Baets. “Dempster–Shafer theory framed in modal logic”. In: *International journal of approximate reasoning* 21.2 (1999), pp. 157–175.

[TBB00] E. Tsiporkova, B. De Baets, and V. Boeva. “Evidence theory in multivalued models of modal logic”. In: *Journal of Applied Non-Classical Logics* 10.1 (2000), pp. 55–81.

[BL04] Ronald Brachman and Hector Levesque. *Knowledge representation and reasoning*. Elsevier, 2004.

[Dem08] Arthur P. Dempster. “Upper and Lower Probabilities Induced by a Multivalued Mapping”. In: *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Ed. by Roland R. Yager and Liping Liu. Springer Berlin Heidelberg, 2008, pp. 57–72.

[GLG09] Artur S d’Avila Garcez, Luis C Lamb, and Dov M Gabbay. *Neural-symbolic cognitive reasoning*. Springer, 2009.

[Bes+17] Tarek R. Besold et al. “Neural-Symbolic Learning and Reasoning: A Survey and Interpretation”. In: *ArXiv* abs/1711.03902 (2017). URL: https://api.semanticscholar.org/CorpusID:1755720.

[LL17] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).

[Mar18] Gary Marcus. “Deep learning: A critical appraisal”. In: *arXiv preprint arXiv:1801.00631* (2018).

[Jai+23] Jitesh Jain et al. “Oneformer: One transformer to rule universal image segmentation”. In: *Proc. of IEEE/CVF Conf. on Comp. Vision and Pattern Recogn.* 2023, pp. 2989–2998.

[Wan+23] Yifeng Wang et al. “Rapid Image Labeling via Neuro-Symbolic Learning”. In: *Proc. of the 29th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. 2023, pp. 2467–2477.

[Zha+23] Zuowei Zhang et al. “Representation of imprecision in deep neural networks for image classification”. In: *IEEE Transactions on NN and Learning Systems* (2023).

[FTB24] Fabian Fingerhut, Elena Tsiporkova, and Veselka Boeva. "Interpretable Data-Driven Risk Assessment in Support of Predictive Maintenance of a Large Portfolio of Industrial Vehicles". In: *2024 IEEE International Conference on Big Data (BigData)*. IEEE. 2024, pp. 2870–2879.

[Hog+24] Eduard Hogea et al. "LogicLSTM: Logically-driven long short-term memory model for fault diagnosis in gearboxes". In: *Journal of Manufacturing Systems* 77 (2024), pp. 892–902.

[Yan+24] Lihe Yang et al. "Depth Anything V2". In: *arXiv preprint arXiv:2406.09414* (2024).

[ZW24] Jie Zhong and Zhenkan Wang. "Implementing deep learning models for imminent component x failures prediction in heavy-duty scania trucks". In: *International Symposium on Intelligent Data Analysis*. Springer. 2024, pp. 268–276.

[DMT25] Michiel Dhont, Adrian Munteanu, and Elena Tsiporkova. "Forecasting Traffic Progression in Terms of Semantically Interpretable States by Exploring Multiple Data Representations". In: *IEEE Transactions on Intelligent Transportation Systems* (2025).

[Fit25] Ricardo Fitas. "Neuro-Symbolic AI for Advanced Signal and Image Processing: A Review of Recent Trends and Future Directions". In: *IEEE Access* (2025).

[Kha+25] Zahra Kharazian et al. "Scania component x dataset: A real-world multivariate time series dataset for predictive maintenance". In: *Scientific Data* 12.1 (2025), p. 493.

[LWT25] Baoyu Liang, Yuchen Wang, and Chao Tong. "AI Reasoning in Deep Learning Era: From Symbolic AI to Neural–Symbolic AI". In: *Mathematics* 13.11 (2025), p. 1707.

[Lu+25] Qiuhao Lu et al. "Explainable diagnosis prediction through neuro-symbolic integration". In: *AMIA Summits on Translational Science Proceedings* 2025 (2025), p. 332.

[MBT25] Giulia Murtas, Veselka Boeva, and Elena Tsiporkova. "An evidence-based neuro-symbolic framework for ambiguous image scene classification". In: *19th International Conference on Neurosymbolic Learning and Reasoning*. 2025. URL: https://openreview.net/forum?id=6UnuZcQ2zY.

[Wan+25] Yang Wang et al. "Learning Reliable and Intuitive Temporal Logic Rules for Interpretable Time Series Classification". In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025, pp. 3067–3078.