Out-of-Distribution Detection with Logical Reasoning

Konstantin Kirchheim, Tim Gonschorek, Frank Ortmeier
Department of Computer Science
Otto von Guericke University Magdeburg, Germany
{firstname}.{lastname}@ovgu.de

Abstract

Machine Learning models often only generalize reliably to samples from the training distribution. Consequentially, detecting when input data is out-of-distribution (OOD) is crucial, especially in safety-critical applications. Current OOD detection methods, however, tend to be domain agnostic and often fail to incorporate valuable prior knowledge about the structure of the training distribution. To address this limitation, we introduce a novel, hybrid OOD detection algorithm that combines a deep learning-based perception system with a first-order logic-based knowledge representation. A logical reasoning system uses this knowledge base at run-time to infer whether inputs are consistent with prior knowledge about the training distribution. In contrast to purely neural systems, the structured knowledge representation allows humans to inspect and modify the rules that govern the OOD detectors' behavior. This not only enhances performance but also fosters a level of explainability that is particularly beneficial in safety-critical contexts. We demonstrate the effectiveness of our method through experiments on several datasets and discuss advantages and limitations. Our code is available online.[1]

1. Introduction

In recent years, Deep Neural Networks (DNN) [36, 27] outperformed classical machine learning models on virtually every task involving large amounts of high-dimensional data, like Natural Language Processing [34] and Computer Vision [13]. Their superior performance enables novel use cases, and current research explores applications in potentially safety-critical areas, such as autonomous vehicles [11] and healthcare [8]. However, DNNs have been shown to only generalize well to new data points as long as they stem from the distribution for which the models have been optimized and tend to make egregiously wrong predictions with high confidence when applied to inputs that are out-of-distribution (OOD) [10, 32]. In safety-critical applications, such OOD inputs pose a safety risk and have to be detected in order to avoid critical errors. However, current OOD detectors are opaque and do not provide an efficient way to incorporate prior knowledge about the target domain. Furthermore, while DNNs often perform well empirically on a surface level, they still exhibit a lack of abstract reasoning abilities, which, according to recent studies, is unlikely to be solved by scaling current architectures [16].

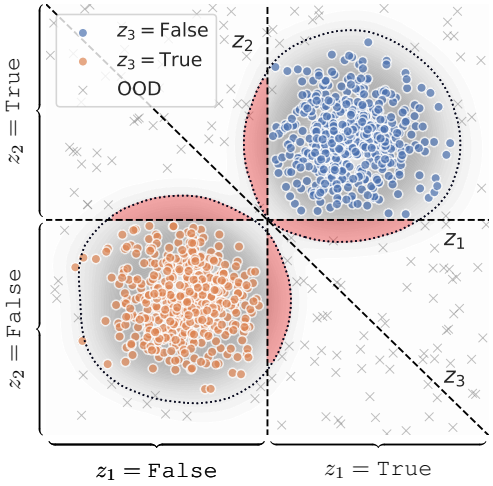


Figure 1: Samples from a joint distribution $p_{\text{data}}(x, y)$ with decision boundaries for OOD detector (dotted) and binary, linear concept detectors (dashed). Logical constraints such as $\neg z_3 \rightarrow z_1 \wedge z_2$ can reject inputs from regions that have no support *a priori*, thereby tightening the decision boundary and providing explanations for the rejected points.

In this work, we aim to address these shortcomings by making the following contributions:

i) We propose a hybrid OOD detection method that identifies OOD inputs by reasoning over a first-order logic knowledge base.

ii) We empirically demonstrate that the presented method

[1] https://github.com/kkirchheim/logic-ood

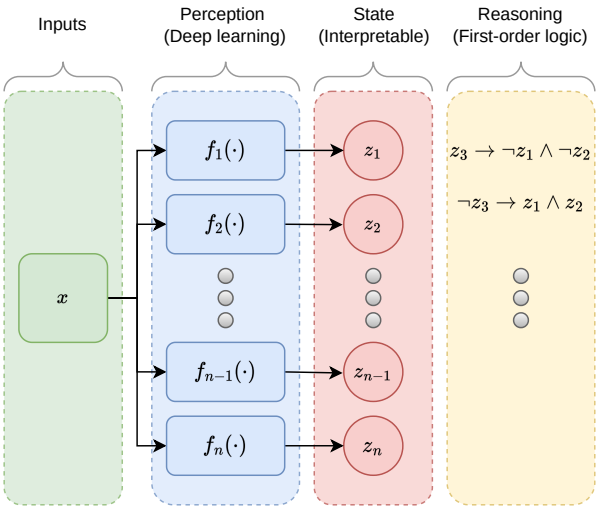


Figure 2: Architecture of the proposed OOD detection framework: Inputs $\mathbf{x}$ are processed by a perception system that detects the presence of semantically meaningful concepts z_i. A logical reasoning system is used to draw conclusions regarding the consistency of the detected concepts with prior domain knowledge encoded in a knowledge base.

can outperform state-of-the-art OOD detectors on some datasets.

iii) We demonstrate that our hybrid design provides advantages over purely neural systems, such as increased explainability and modularity.

2. Background

Let p_{data} be the data-generating distribution. The set of OOD inputs can then be defined as $\mathcal{X}_O = \{\mathbf{x} \in \mathcal{X} : p_{\text{data}}(\mathbf{x}) < \alpha\}$, where $\alpha \in [0, 1]$ is a threshold. An OOD detector $D_f(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}$ can be defined as a function that assigns outlier scores to inputs $\mathbf{x} \in \mathcal{X}$, such that $D_f(\mathbf{x})$ is high if $\mathbf{x}$ is more likely to be OOD. Here, f is some DNN with input space $\mathcal{X}$. Outliers are then identified by applying a threshold τ such that

$$\text{outlier}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad D_f(\mathbf{x}) \geq \tau \\ 0 & \text{else} \end{cases} . \tag{1}$$

In recent years, various methods for OOD detection have been proposed [45], most of which are based on either the posterior class membership probabilities predicted by a DNN-based classifier [17, 9], the unnormalized logits [1, 30, 14], or the representation of the input in some higher layer [43, 28, 35]. For example, the Maximum Softmax Probability (MSP) baseline method [17] is defined as

$$D_f(\mathbf{x}) = -\max_y q_\theta(y|\mathbf{x}) \tag{2}$$

$$= -\max_y \frac{\exp(f_\theta(\mathbf{x})_y / T)}{\sum_{i=1}^{N} \exp(f_\theta(\mathbf{x})_i / T)}, \tag{3}$$

where $q_\theta(y|\mathbf{x})$ is the posterior probability the DNN f with parameters θ assigns to class y, and $f(\cdot)_y$ refers to the y^{th} output of f [17]. T, which is 1 by default, is an optional temperature value that can be used to scale the logits to improve the model's calibration [12]. More recently, activation pruning methods were proposed that aim to rectify unusual activations in some layers of the model [39, 7, 40].

3. Proposed Framework

State-of-the-art OOD detectors, as outlined above, are based on statistics of the activations of neurons while neglecting the semantics of these neurons in the context of the task. We argue that the integration of priors in the form of explicit rules on semantic concepts of the domain, designed by and for humans, can increase the performance and explainability of OOD detectors.

As a motivating example, consider Fig. 1: The image depicts samples from a joint distribution $p_{\text{data}}(\mathbf{x}, y)$ as well as the OOD decision boundary $D(\mathbf{x}) = \tau$ of some detector D. The set of inputs rejected by D is $\mathcal{X}_R = \{D(\mathbf{x}) \geq \tau\}$. Let us assume that we know *a priori* that points from the *blue* class can only reside in the top-right quadrant, while points from the *orange* class can only lie in the bottom-left quadrant. These constraints define a set of OOD points $\mathcal{X}_P \subseteq \{\mathbf{x} \in \mathcal{X} : p_{\text{data}}(\mathbf{x}) = 0\} \subseteq \mathcal{X}_O$. Now let $\mathcal{X}'_R = \mathcal{X}_R \cup \mathcal{X}_P$. Clearly $|\mathcal{X}_O \cap \mathcal{X}'_R| \geq |\mathcal{X}_O \cap \mathcal{X}_R|$, which implies that rejecting points from $\mathcal{X}_P$ can only improve the OOD detection performance since the additionally rejected points $\mathcal{X}_P \setminus \mathcal{X}_R$ (red shaded regions) will by construction be OOD.

In this work, we propose a framework for OOD detection that allows integrating priors about which observations are considered possible inside of a certain domain. The proposed framework, whose architecture is depicted in Fig. 2, consists of three components: (1) a *perception system*, possibly based on DNNs, (2) a *state*, which is updated by the perception system, and (3) a *logical reasoning system* that checks if the current state is compatible with a set of constraints. Each of these components will be described in detail in the following.

3.1. Perception System

In practical applications, domain knowledge is often only present in the form of constraints on abstract semantic concepts that are difficult to correlate with raw observations. In other words, constraints are often not defined in the input space $\mathcal{X}$, but on a more abstract level. For example, when classifying traffic signs in images, we have strong priors

regarding the shape and color that certain signs should have and would consider any observation that violates these constraints as highly unlikely. However, how the input pixels relate to the abstract concept of *shape* is not obvious.

The perception system aims to bridge the gap between the high-dimensional sensory inputs and prior knowledge by mapping raw percepts to abstract semantic concepts. We assume that each input $\mathbf{x} \in \mathcal{X}$ is associated with some concepts z_i that each take values from a corresponding set $\mathcal{Z}_i$. Such concepts could include, for example, the class of the input, as well as other properties or attributes, such as its shape, color, degree of rotation, *et cetera*. We refer to the touple $z = \langle z_1 \in \mathcal{Z}_1, ..., z_N \in \mathcal{Z}_N \rangle \in \mathcal{Z}$ associated with some $\mathbf{x}$ as the *true state* of $\mathbf{x}$, where the state space $\mathcal{Z}$ is the cartesian product of all $\mathcal{Z}_i$.

The perception system consists of a set of concept detectors $f_i : \mathcal{X} \to \mathcal{Z}_i$ that detect the presence of certain concepts in a given input. We refer to the tuple of detected concepts $\hat{z} = \phi(\mathbf{x}) = \langle f_1(\mathbf{x}), ..., f_N(\mathbf{x}) \rangle \in \mathcal{Z}$ as the *predicted state* for some input $\mathbf{x}$.

In the following, we parameterize the concept detectors f_i by DNNs. Note, however, that in general, other choices are possible, and for some applications, choosing a different parameterization might prove beneficial. While we could implement concept detectors such that they use a shared backbone with multiple concept detection heads, we suspect that this could lead to correlated predictions for otherwise independent concepts and, therefore, to correlated detection errors, which should be avoided. In the following, we will, therefore, use a different DNN for each concept if not stated otherwise.

3.2. Reasoning System

Based on the state space, we can define a set of constraints on the semantic concepts that restrict the set of *states* that are considered possible for the intended domain. While there are several conceivable approaches to implement such a system, we propose to use a logical reasoning system that operates on constraints in the form of first-order logic formulas defined over the concepts z_i detected by the perception system. These constraints are stored in a knowledge base (KB). The reasoning system implements a reasoning engine $\texttt{sat}_{\texttt{KB}} : \mathcal{Z} \to \{0, 1\}$ that is able to infer whether a given state z satisfies all formulas in the knowledge base, or in other words if z models the KB, i.e., $z \models KB$. Thereby, we effectively partition the state space into those states that satisfy the KB and those that do not. We refer to the states $z \in \mathcal{S} = \{z \in \mathcal{Z} : \texttt{sat}_{\texttt{KB}}(z) = 1\}$ that satisfy all formulas in the KB as *valid states*, and to all other states in $\mathcal{Z} \setminus \mathcal{S}$ as *invalid states*.

For the introductory example in Fig. 1, a simple knowledge base could contain the rule $\neg z_3 \to z_1 \wedge z_2$, which asserts that, if z_3 is false (i.e., the point belongs to the *blue* class), z_1 and z_2 must both be true. If this condition is not satisfied for some state z, the $\texttt{sat}_{\texttt{KB}}$ function evaluates to 0, and the predicted state is considered invalid.

Should a predicted state be invalid, we can mark the corresponding input as OOD. This leads to a naïve approach that assigns an outlier score of 0 if the predicted state satisfies all constraints in the knowledge base and 1 if it does not, or formally:

$$D_\phi(x, \mathcal{S}) = \begin{cases} 0 & \text{if} \quad \phi(\mathbf{x}) \in \mathcal{S} \\ \tau & \text{else} \end{cases} . \tag{4}$$

However, this naïve approach neglects outliers in regions not prohibited by the KB (i.e., $\mathcal{X}_O \setminus \mathcal{X}_P$). Thus, we propose to calculate the outlier score as

$$D'_\phi(\mathbf{x}, \mathcal{S}) = \begin{cases} \sum_{i=1}^{N} \lambda_i D_{f_i}(\mathbf{x}) & \text{if} \quad \phi(\mathbf{x}) \in \mathcal{S} \\ \tau & \text{else} \end{cases} \tag{5}$$

where D_{f_i} is some OOD detector for model f_i and $\lambda_i \in \mathbb{R}$ are weighting coefficients. In other words, the outlier score is a linear combination of the outlier score of the concept detectors f_i, provided that the predicted state is valid and τ (i.e., OOD) otherwise. In this work, we use the MSP baseline detector from Eq. (3) as D_{f_i}.

3.3. Theoretical Properties

The described method has a number of theoretical properties, proofs of which are provided in the supplementaries. First, our detector is a generalization of existing OOD detectors in the following sense:

Proposition 1. *For some detector D_f, there exists D'_ϕ, λ and $\mathcal{S}$ such that $\forall \boldsymbol{x} : D'_\phi(\boldsymbol{x}, \mathcal{S}) = D_f(\boldsymbol{x})$.*

Computational Complexity Computing $\phi(\mathbf{x})$ scales linearly with the number of concept detectors. However, when the concept detectors are independent, as proposed, the computation of the individual concepts $f_1(\mathbf{x}), ..., f_N(\mathbf{x})$ can be trivially parallelized, implying that the overall response time can remain constant. Inference in FOL, in general, is undecidable; however, there are subsets for which efficient inference algorithms exist, such as Horn clauses, for which checking if a particular state violates a constraint can be done in polynomial time. While, for Horn clauses, it would, in theory, be possible to enumerate all valid states and implement the reasoning system as a look-up table, enumerating all valid states quickly becomes intractable:

Proposition 2. *Enumerating the valid states $\mathcal{S}$ is #P-Hard.*

However, for small state spaces, precomputing all valid states to accelerate inference can be feasible.

Modularity Modifying deep learning systems usually requires re-training since the knowledge is encoded in the model's parameters, and it is difficult to determine how the parameters have to be adjusted to cause a certain change in behavior. Our framework, however, allows adjusting the system easily. This provides advantages over existing OOD detectors, as the system can be constructed incrementally and can be customized to meet evolving requirements.

For instance, we can add a concept detector f' to the perception system ϕ, resulting in an extended perception system ϕ' with state $z' \in \mathcal{Z}'$.

Proposition 3. *When adding a concept detector to ϕ, the following holds: $p(\phi'(\boldsymbol{x}) = z' \mid \boldsymbol{x}) \leq p(\phi(\boldsymbol{x}) = z \mid \boldsymbol{x})$.*

Thus, when increasing the number of concept detectors, it becomes increasingly likely that the predicted state is not equal to the true state. Similarly, we can add a new constraint to the KB, which leads to a new set of valid states $\mathcal{S}'$.

Proposition 4. *Adding a constraint to a KB with valid states $\mathcal{S}$ results in a new set of valid states $\mathcal{S}'$ such that $\mathcal{S}' \subseteq \mathcal{S}$.*

It follows that adding a constraint can not increase the probability that an input is marked as IN. Under reasonable assumptions, it can be shown that adding additional concepts and constraints to the system will render it increasingly unusable but not increasingly unsafe, as the system will tend to reject all inputs. Intuitively, the reason is that adding concepts and constraints to the system will never cause an input that was marked as OOD by the reasoning system before to be marked as IN afterward.

4. Experimental Evaluation

In the following, we validate our approach on three datasets that we selected because they readily provide prior knowledge that can be utilized. The implementation of the perception system is based on PyTorch [33] and PyTorch-OOD [23], while the knowledge base and the reasoning system are implemented in Prolog.

Several previous works demonstrated that the results of experiments involving DNNs, including in the domain of out-of-distribution detection, can vary significantly with the random seed [4, 22, 24]. To account for this variation in experimental outcomes, we averaged the results over 10 trials with different random seeds and tested our results for statistical significance.

4.1. Ablations

To demonstrate the effect of each component of our proposed architecture, we conduct ablation studies with the following variants:

Logic To validate the general concept, we provide results for naïve detector that only verifies if the predicted state is valid, as formulated in Eq. (4).

Ensemble As a baseline, we calculate the outlier score as the mean of the baseline MSP outlier scores, as given by Eq. (3), over all of the concept detectors $\frac{1}{N}\sum_{i=1}^{N} D_{f_i}(\mathbf{x})$. This can be seen as a version of our approach where the knowledge base does not impose any restrictions, and all states are valid, such that $\mathcal{S} = \mathcal{Z}$. Compared to the usual ensemble approach [26], where models are optimized from different initializations, in this approach, the models are optimized to predict different targets.

LogicOOD To demonstrate the effect of the KB, we provide results for our approach with a KB where the outlier score is calculated as Eq. (5). Note that this method does not require example outliers during training.

LogicOOD+ To demonstrate the modularity of our framework, we extend our method by an additional concept detector that is trained with example outliers to distinguish between IN and OOD. We also add a rule to assert that all inputs should be IN. We expect this method to outperform LogicOOD due to the additional outliers used, as incorporating example outliers into the training is known to increase the detection performance [19, 25, 21]. Note that the rest of the system requires no change.

T-LogicOOD We furthermore include results for temperature-calibrated concept detectors [12]. Optimal calibration parameters are estimated on a validation set. Note that temperature scaling does not affect the results of the Logic approach, as it does not change $\arg\max_y q_\theta(y \mid \mathbf{x})$.

4.2. German Traffic Sign Recognition

Correctly detecting and classifying traffic signs is an essential application for autonomous driving. The German Traffic Sign Recognition Benchmark (GTSRB) [37] dataset contains images of German traffic signs from 43 different classes, captured in natural environments and under diverse illuminations, weather conditions, and from different angles and with partial occlusions. In our first experiment, we consider the GTSRB images as in-distribution.

4.2.1 Knowledge Base

It is a common phenomenon that objects in our environment are associated with particular attributes. For example, traffic regulations govern the shape and color of traffic signs. For the GTSRB, we populate the KB with formulas representing these laws. In this setting, we consider the three concepts

label, color, and *shape*, where *label* refers to the class of a traffic sign. Each of these concepts is detected by a different concept detector $f_{\mathcal{L}}$, $f_{\mathcal{C}}$, and $f_{\mathcal{S}}$, respectively. The KB then consists of rules in the form

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{stopsign}) \rightarrow \texttt{has\_color}(\hat{z}, \texttt{red}) \wedge \texttt{has\_shape}(\hat{z}, \texttt{octagon}) \tag{6}$$

which specifies that, if something is labeled as *stop-sign*, it also has to be a *red octagon*. Thus, the state $\langle \texttt{stopsign}, \texttt{octagon}, \texttt{red} \rangle$ satisfies this constraint. For each traffic sign, we generate rules that associate the traffic signs with a unique combination of color and shape.

In this setting, the entire state space contains 1720 possible states. However, the KB restricts the set of valid states to $|\mathcal{S}| = 43$, one for each sign type. The complete KB is provided with the code.

For LogicOOD+, we can extend our method by an additional concept detector $f_{\mathcal{T}}$ that is trained to decide if an image contains a traffic sign or not, using a set of available training outliers. We can then extend the KB with the rule

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{traffic\_sign}) \tag{7}$$

which asserts that all IN images must depict a traffic sign.

4.2.2 Data

The GTSRB dataset itself features 51,840 training and 12,630 test images from 43 classes. As OOD data, we use five datasets with unrelated images: Textures [6], LSUN Crop [29], LSUN Resize [29], TinyImageNet Crop [29] and TinyImageNet Resize [29]. As training outliers for LogicOOD+, we use a cleaned subset of the 80 million tiny images database [41, 2], which has no overlap with the test outliers.

4.2.3 Results

Results are provided in Tab. 1. The Ensemble approach outperforms most other methods on its own, which suggests that considering several targets instead of one already increases performance on this dataset. Furthermore, we see that LogicOOD significantly (unequal variances t-test: $p < 0.05$) outperforms other OOD detection approaches based on probabilities, logits, features, and activation pruning, as well as the Ensemble method, which demonstrates the effect of the reasoning system. Confidence calibration has no significant impact on the results.

4.3. PrimateNet

Instead of constructing domain-specific knowledge bases from the ground up, we can also use existing knowledge graphs. The large-scale ImageNet database contains images from more than 21,000 classes, where each class is associated with a concept in the WordNet [31] knowledge base. By defining *is-a* or *subtype* relations between different concepts, such as *"a mammal is an animal"*, WordNet structures concepts into a hierarchy. PrimateNet is a subset of the ImageNet that contains all classes in the subtree of the WordNet hierarchy, beginning with *Primate* as the root node.

4.3.1 Knowledge Base

To populate the knowledge base for PrimateNet, we convert the hierarchical *is-a* relations that exist between different classes in the PrimateNet dataset into a set of first-order logic statements, such as

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{lesser\_ape}) \rightarrow \texttt{is\_a}(\hat{z}, \texttt{ape}) \tag{8}$$

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{gibbon}) \rightarrow \texttt{is\_a}(\hat{z}, \texttt{lesser\_ape}) \tag{9}$$

which assert that each lesser ape is an ape, and each gibbon is also a lesser ape. Additionally, we add rules that assert that concepts at the same level of the hierarchy are mutually exclusive, such as

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{ape}) \rightarrow \neg \texttt{is\_a}(\hat{z}, \texttt{monkey}) \,. \tag{10}$$

However, we could also encode such rules directly in the DNN structure by using a multi-class classifier for each level of the hierarchy instead of a binary classifier for each node. From the rules (8) and (9), it follows logically that gibbons must also be apes, and, together with rule (10), that gibbons can not be monkeys. Overall, these constraints partition the state space, which contains 2048 states in total, into 16 valid and 2032 invalid states.

The perception system contains a concept detector for the *class*, which discriminates between the 16 ImageNet classes in the leaf nodes of the hierarchy, and 7 binary classifiers for the intermediate nodes (except the root node *Primate*, since we can not train a discriminative model for this concept without example outliers) so that the predicted state consists of 8 variables.

For LogicOOD+, we leverage example outliers by training a discriminative detector that decides whether an input image contains a primate. For each of *monkey*, *lemur* and *ape*, we then add a rule in the form

$$\forall \hat{z} : \texttt{is\_a}(\hat{z}, \texttt{ape}) \rightarrow \texttt{is\_a}(\hat{z}, \texttt{primate}) \,. \tag{11}$$

It follows that all images in the PrimateNet belong to the *primate* class. Therefore, each of the IN distribution images must depict a primate, and all other images are considered OOD.

4.3.2 Data

The PrimateNet contains images from 16 ImageNet classes, where, for each class, there are 1300 images for training and 50 for testing. We treat all of these images as IN. In our evaluation, we use datasets with images from different datasets as OOD data: Textures [6], ImageNet-O [20], Fooling Images [38], ImageNet-R [15], ImageNet-A [20], NINCO [3] and iNaturalist [42]. We use 10 randomly selected classes from the ImageNet dataset as training outliers for LogicOOD+. These classes do not overlap with the OOD test data. Further details are provided in the supplementary.

4.3.3 Results

Results are provided in Tab. 1. We observe that the Logic-based approaches significantly outperform random guessing but are often outperformed by other methods.

Visualizations of normalized outlier scores for the NINCO (OOD) and PrimateNet (IN) data are provided in Fig. 3. As we can see, the introduction of logical constraints (Fig. 3b) leads to the assignment of high outlier scores to some of the IN samples. Investigating these rejected IN samples, we notice that a large proportion is assigned the wrong PrimateNet class by the class concept detector (46.8% of rejected IN samples are misclassified, compared to an overall error rate of 18.6% on IN images). We also see that including outlier examples (Fig. 3d) causes the system to reject most OOD samples; however, the misclassified IN samples remain rejected. We also observe a slight drop in performance for calibrated models.

We conclude that the reasoning system does, in fact, work as expected; however, the tendency to reject IN images that are assigned the wrong class with high outlier scores decreases performance measures on the OOD detection task.

4.4. Fruit Recognition

We additionally conduct experiments on a dataset with images of 33 different types of fruit. We only use two concept detectors: one that predicts the type of the fruit and one that predicts its color. We then verify that the detected color indeed matches the detected fruit (e.g., a *banana* should be *yellow*) by using a very simple KB. The remainder of the setup is similar to the GTSRB experiments. As we can see in Tab. 1, LogicOOD achieves competitive performance and is only outperformed by feature-based detectors like ViM and Mahalanobis, which suggests that the performance could be further increased by using a feature-based OOD detector. Statistically, there is no significant difference in the performance of LogicOOD+ and ViM ($p > 0.05$). Again, temperature scaling has no significant effect.

5. Discussion

5.1. Advantages

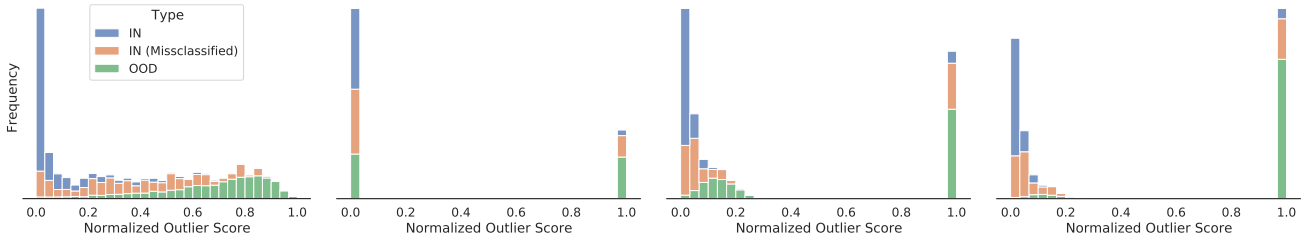
Explainability Explainability is generally considered a desirable property of systems, particularly in safety-critical applications. Current OOD detection methods predict a scalar outlier score that provides no further insights into the causes leading to this score. In contrast to DNNs, the logical reasoning system provides our system with a certain degree of interpretability in the sense that we are able to provide meaningful and intuitive explanations based on human-interpretable concepts for many of the decisions of the system. We will illustrate this in the following using several IN and OOD samples from the GTSRB, provided in Fig. 4.

Fig. 4a depicts an outlier that was classified as a *Priority Road* sign with 100% confidence. However, the concept detectors detected a red square in the image, while we would expect a yellow square. Contrary to a scalar confidence value, we can now provide a justification for *why* the prediction of the classifier should not be trusted in this case, despite its high confidence: to the perception system, the image looks like a red square, which contradicts the rules encoded in our knowledge base.

Fig. 4b depicts an occluded IN image that was missclassified as a *Keep Right* sign with 96.35% confidence. The concept detectors, however, correctly identify the shape and color of the traffic sign, which do not match the predicted label since we would expect the sign to be a *blue circle*. This example illustrates that the consistency-enforcing behavior is also able to detect certain types of misclassifications, which, however, negatively impacts the OOD detection performance measures.

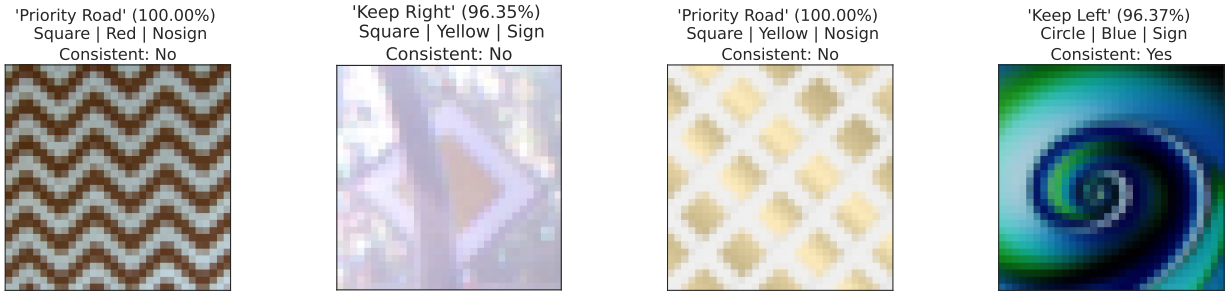
Fig. 4c depicts an outlier that was classified as a *Priority Road* sign with 100% confidence. The concept detectors perceive a *square* shape and the color *yellow*, which is consistent with the domain knowledge about *Priority Road* signs. However, in this case, the binary sign-concept detector of LogicOOD+ determines that this input is not an actual traffic sign, which, again, provides us with an explanation: the shape and color in the image match the predicted sign, while the image does not depict a traffic sign.

Finally, Fig. 4d depicts an outlier that was classified as a *Keep Left* sign with 96.37% confidence. The concept detectors identify a *blue circle*, which is consistent with the domain knowledge. In this case, the sign detector mistakenly marks the image as a real traffic sign. Inspecting the image, we intuitively see that the color and shape prediction seem reasonable. Hence, while the system failed, we can provide an explanation: to the perception system, the image looked like an actual blue traffic sign in the shape of a circle, closely resembling a roundabout sign.



(a) MSP (b) Logic (c) LogicOOD (d) LogicOOD+

Figure 3: Frequencies of normalized outlier scores for the MSP baseline method and our proposed approaches for the PrimateNet as IN and NINCO [3] as OOD data. Logic-based methods reject some of the in-distribution inputs, which we attribute to the inaccuracy of the concept detectors.



(a) OOD (b) IN (c) OOD (d) OOD

Figure 4: Examples of IN and OOD samples with corresponding confidence values as predicted by the MSP baseline method. Figs. 4a to 4c result in states that are inconsistent with the knowledge base and therefore marked as OOD.

Completeness of Rules Another advantage of our hybrid system, compared to purely neural approaches, is the transparency obtained by incorporating a set of explicit, human-understandable rules, which allows us to reason about the system's properties by assessing the knowledge base. For example, a safety assessor can reason about the completeness of the (e.g., formulas in the) knowledge base, which constitutes an important step in safety evaluations. In contrast, an inspection of the weights of a DNN does currently not provide much insight into its safety-related properties.

5.2. Limitations

Scalability As the concept detectors are usually imperfect, adding an additional concept detector to the system increases the probability that the predicted state differs from the true state (Proposition 3), which could lead to rejected IN inputs. As evident from the PrimateNet example, this property can severely impact performance measures. However, as we outlined in Sec. 3.3, scaling is likely to lead to many rejected inliers and not to wrongly accepted outliers. This is also supported by our experiments.

Additional Labeling Manually annotating a large number of concepts for individual images in a dataset can be tedious. However, while it may seem that our approach requires additional labeling, we demonstrated that additional labels can often be derived from prior knowledge. For example, the shape and color of a traffic sign are known *a priori*; that is, they are known without having to inspect a particular instance of the sign, which makes manual labeling unnecessary.

Continuous Concepts To handle continuous concepts such as $\mathcal{Z}_i \subseteq \mathbb{R}$ for some $i \in \mathbb{N}$ and to, for example, assert that the value must lie in a specific interval, one would have to extend the presented approach to real numbers. However, we consider this a rather theoretical limitation since, in practice, reasoning systems like Prolog provide native support for such operations, and the implementation is straightforward.

6. Conclusion & Future Work

In this work, we introduced a method for OOD detection that allows integrating prior knowledge in the form of first-order logic constraints to infer whether a given input is OOD. We found that the proposed approach can outperform state-of-the-art methods in some cases while being more transparent and modular overall, which is an important

Table 1: Mean performance and corresponding standard error of different OOD detection methods in our experiments. Results averaged over 10 experimental trials with different random seeds and different datasets. The best result is in bold, and the second best is underlined. ↑ indicates that higher values are better, while ↓ indicates the opposite – all values in percent.

Detector	AUROC ↑		AUPR-IN ↑		AUPR-OUT ↑		FPR95 ↓	
	GTSRB (WideResNet [46] + ImageNet 1K Pre-Training [18])							
MSP [17]	99.04	± 0.07	98.35	± 0.14	99.29	± 0.05	2.54	± 0.15
EBO [30]	99.03	± 0.11	98.76	± 0.14	99.08	± 0.12	2.26	± 0.27
MaxLogit [14]	99.01	± 0.11	98.73	± 0.14	99.07	± 0.12	2.29	± 0.27
Entropy [5]	99.15	± 0.07	98.64	± 0.13	99.33	± 0.06	2.46	± 0.15
ReAct [39]	99.04	± 0.10	98.77	± 0.13	99.08	± 0.12	2.21	± 0.24
Mahalanobis [28]	99.70	± 0.02	99.40	± 0.06	99.83	± 0.01	1.11	± 0.05
ViM [43]	96.96	± 0.08	95.95	± 0.10	99.75	± 0.02	6.08	± 0.16
Ensemble [26]	99.77	± 0.03	99.58	± 0.05	99.86	± 0.01	0.99	± 0.07
Logic (ours)	86.08	± 0.91	91.76	± 0.54	91.76	± 0.45	100.00	± 0.00
LogicOOD (ours)	<u>99.85</u>	± 0.01	<u>99.74</u>	± 0.02	<u>99.92</u>	± 0.01	<u>0.60</u>	± 0.04
Logic+ (ours)	99.92	± 0.01	99.90	± 0.01	99.97	± 0.00	0.13	± 0.01
LogicOOD+ (ours)	**99.94**	± 0.01	**99.91**	± 0.01	**99.97**	± 0.00	**0.13**	± 0.01
T-LogicOOD (ours)	<u>99.85</u>	± 0.01	<u>99.74</u>	± 0.02	<u>99.92</u>	± 0.01	<u>0.60</u>	± 0.04
T-LogicOOD+ (ours)	**99.94**	± 0.01	**99.91**	± 0.01	**99.97**	± 0.00	**0.13**	± 0.01
	PrimateNet (ResNet-50 [44] + ImageNet 1K Pre-Training [18])							
MSP [17]	94.95	± 0.10	99.18	± 0.02	78.00	± 0.34	22.13	± 0.34
EBO [30]	97.39	± 0.07	99.61	± 0.01	84.53	± 0.33	12.32	± 0.34
MaxLogit [14]	97.15	± 0.06	99.56	± 0.01	84.21	± 0.32	12.79	± 0.32
Entropy [5]	96.46	± 0.07	99.46	± 0.01	81.05	± 0.34	16.47	± 0.41
ReAct [39]	<u>98.88</u>	± 0.04	<u>99.85</u>	± 0.01	<u>92.76</u>	± 0.18	<u>5.56</u>	± 0.15
Mahalanobis [28]	98.18	± 0.06	99.77	± 0.01	91.87	± 0.20	7.69	± 0.23
ViM [43]	**99.59**	± 0.02	**99.95**	± 0.00	**97.46**	± 0.04	**1.74**	± 0.06
Ensemble [26]	95.89	± 0.08	99.19	± 0.03	85.37	± 0.17	14.12	± 0.20
Logic (ours)	73.90	± 0.30	96.53	± 0.05	56.53	± 0.13	100.00	± 0.00
LogicOOD (ours)	92.19	± 0.09	98.54	± 0.02	83.47	± 0.19	15.46	± 0.23
Logic+ (ours)	89.45	± 0.13	98.75	± 0.02	73.81	± 0.32	55.94	± 3.45
LogicOOD+ (ours)	94.35	± 0.07	99.19	± 0.01	88.69	± 0.14	10.89	± 0.13
T-LogicOOD (ours)	91.72	± 0.09	98.48	± 0.02	80.72	± 0.21	19.21	± 0.29
T-LogicOOD+ (ours)	94.25	± 0.07	99.18	± 0.01	87.46	± 0.18	11.37	± 0.17
	Fruits (WideResNet [46] + ImageNet 1K Pre-Training [18])							
MSP [17]	96.40	± 0.59	99.13	± 0.14	85.99	± 1.79	18.35	± 2.84
EBO [30]	96.74	± 0.52	99.24	± 0.12	86.15	± 1.70	18.55	± 2.77
MaxLogit [14]	96.73	± 0.52	99.24	± 0.12	86.14	± 1.69	18.56	± 2.77
Entropy [5]	96.61	± 0.56	99.20	± 0.13	86.22	± 1.75	18.09	± 2.79
ReAct [39]	88.21	± 1.13	96.94	± 0.34	63.33	± 2.01	50.98	± 2.25
Mahalanobis [28]	99.86	± 0.05	99.97	± 0.01	99.29	± 0.24	0.36	± 0.17
ViM [43]	**99.94**	± 0.02	**99.99**	± 0.00	**99.72**	± 0.09	**0.06**	± 0.02
Ensemble [26]	98.19	± 0.33	99.54	± 0.09	93.54	± 0.96	8.58	± 1.45
Logic (ours)	74.55	± 1.07	95.50	± 0.18	64.78	± 0.47	100.00	± 0.00
LogicOOD (ours)	98.51	± 0.26	99.63	± 0.07	94.23	± 0.88	7.33	± 1.34
Logic+ (ours)	99.88	± 0.04	99.97	± 0.01	99.83	± 0.04	<u>0.17</u>	± 0.06
LogicOOD+ (ours)	<u>99.91</u>	± 0.03	<u>99.98</u>	± 0.01	<u>99.93</u>	± 0.03	<u>0.17</u>	± 0.06
T-LogicOOD (ours)	98.50	± 0.27	99.62	± 0.07	94.18	± 0.93	7.22	± 1.37
T-LogicOOD+ (ours)	<u>99.91</u>	± 0.03	<u>99.98</u>	± 0.01	99.91	± 0.03	<u>0.17</u>	± 0.06

aspect in safety-critical applications.

While we demonstrated that leveraging domain-specific priors can provide state-of-the-art OOD detection performance in some settings, strict logical reasoning may sometimes be too inflexible for real-world applications, as it does not account for the uncertainty in the predictions of the perception system. Consequently, supplementing the system with probabilistic reasoning capabilities appears to be a promising avenue for future work.

Acknowledgements We acknowledge funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK; grant agreement 19I21039A).

References

[1] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[2] Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1536–1546. IEEE, 2021.

[3] Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing imagenet out-of-distribution detection evaluation. *arXiv preprint arXiv:2306.00826*, 2023.

[4] Xavier Bouthillier, César Laurent, and Pascal Vincent. Unreproducible research is reproducible. In *International Conference on Machine Learning*, pages 725–734, 2019.

[5] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 5128–5137, 2021.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[7] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. *arXiv preprint arXiv:2209.09858*, 2022.

[8] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[10] Ben Goertzel, Yehezkel S. Resheff, Itay Lieder, and Tom Hope. All together now! the benefits of adaptively fusing pre-trained deep representations. In *International Conference on Artificial General Intelligence*. Springer, Feb. 2019.

[11] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[14] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, 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*, pages 8340–8349, 2021.

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*, 2017.

[18] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018.

[20] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[21] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. Multi-class hypersphere anomaly detection. In *Proceedings of the 26th International Conference for Pattern Recognition*, August 2022.

[22] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. On challenges for the reproducibility in deep anomaly detection. In *ICPR workshop on Reproducible Research in Pattern Recognition*, 2022.

[23] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. PyTorch-OOD: A library for out-of-distribution detection based on pytorch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4351–4360, June 2022.

[24] Konstantin Kirchheim, Tim Gonschorek, and Frank Ortmeier. Addressing randomness in evaluation protocols for out-of-distribution detection. *2nd Workshop on Artificial Intelligence for Anomalies and Novelties at IJCAI*, 2021.

[25] Konstantin Kirchheim and Frank Ortmeier. On outlier exposure with generative models. In *NeurIPS Machine Learning Safety Workshop*, December 2022.

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

[29] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks.

In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[30] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[32] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[35] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

[36] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[37] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[38] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2029, 2019.

[39] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021.

[40] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, page 691–708. Springer, 2022.

[41] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

[42] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[43] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4921–4930, 2022.

[44] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

[45] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *ArXiv*, abs/2110.11334, 2021.

[46] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.