Attack logics, not outputs: Towards efficient robustification of deep neural networks by falsifying concept-based properties

Raik Dankworth[1], Gesina Schwalbe[1]

[1]*University of Lübeck, Germany*

Abstract

Deep neural networks (NNs) for computer vision are vulnerable to adversarial attacks, i.e., miniscule malicious changes to inputs may induce unintuitive outputs. One key approach to verify and mitigate such robustness issues is to falsify expected output behavior. This allows, e.g., to locally proof security, or to (re)train NNs on obtained adversarial input examples. Due to the black-box nature of NNs, current attacks only falsify a class of the *final output*, such as flipping from `stop_sign` to `¬stop_sign`. In this short position paper we generalize this to search for generally *illogical* behavior, as considered in NN verification: falsify constraints (*concept-based properties*) involving further human-interpretable concepts, like `red ∧ octogonal → stop_sign`. For this, an easy implementation of concept-based properties on already trained NNs is proposed using techniques from explainable artificial intelligence. Further, we sketch the theoretical proof that attacks on concept-based properties are expected to have a reduced search space compared to simple class falsification, whilst arguably be more aligned with intuitive robustness targets. As an outlook to this work in progress we hypothesize that this approach has potential to efficiently and simultaneously improve logical compliance and robustness.

Keywords

Trustworthy AI, Neural Network Verification, Adversarial Attack, Explainable Neural Network, Concept-based XAI, Computer Vision

1. Introduction

Neural Networks (NNs) excel in processing subsymbolic inputs like images, and are increasingly being considered for use in safety-critical domains [1]. This makes it crucial to ensure their robust and intuitive generalization, at least around known training cases. One tool to evaluate vulnerability to malicious attacks are Adversarial Attacks (AAs): These craft inputs that induce incorrect or unexpected predictions, using minimal modifications to a correctly handled input $\mathbf{x}$ with $\mathbf{y} = f(\mathbf{x})$ [2, 3, 4, 5, 6, 7, 8]. However, existing attacks solely focus on altering the model's final output, i.e., falsify $\forall \mathbf{x}' \in \text{Nbhd}(\mathbf{x}): f(\mathbf{x}') = y$ for some neighborhood $\text{Nbhd}(\mathbf{x})$ around $\mathbf{x}$ like an ϵ-ball. This disregards whether the prediction still conforms to high-level, interpretable properties. Common examples of known properties are sufficient conditions, e.g., $\text{red}(\mathbf{x}) \wedge \text{octogonal}(\mathbf{x}) \implies \text{stop\_sign}(\mathbf{x})$ in traffic sign recognition from images x; and necessary conditions, like $\neg\text{octogonal}(\mathbf{x}) \implies \neg\text{stop\_sign}(\mathbf{x})$. More general, rules involving unary predicates not available from the NN outputs are here called *concept-based properties*. Rich semantic rules are known to be well suited for runtime plausibility monitoring [9, 10] and respective fixing of NN outputs [10, 11, 12]. In particular, they don't constrain the local *output* to be correct, but the underlying *general logical reasoning* locally around the sample.

One reason why falsification of such informative constraints are not considered for attack generation is that they require outputs for all involved predicates—not only the available final output, like `stop_sign`. These however, might need a considerable amount of training data or hyperparameter tuning if added right away during the training; or, even worse, not all properties and thus not all required concepts might be known at training time due to specification gaps or later domain transfer.

The trick we now use here is that NNs automatically learn to encode task-related concepts in their intermediate outputs. For example, when trained for `stop_sign` recognition, the NN may implicitly learn to identify `octagons`, `red color`, and the `stop_label`. Post-hoc supervised concept-based explainability methods [13, 14, 15, 16] can recover this information in a very sample-efficient manner with minimal additions to the NN structure.

Altogether, we propose and theoretically analyze a general AA goal —the *Concept-based Property Attack (ConPAtt)*—that explicitly targets falsification of symbolic *concept-based properties* over non-symbolic inputs. As we will show, our formulation offers a more general way to define both targeted and untargeted attacks. Furthermore, as opposed to classical attacks that purely change the output, our attack on ¬`octogonal` $\implies$ ¬`stop_sign` can produce an image still classified as `stop_sign`, in which the `octogonal` concept is no longer recognized. This newly allows to uncover failure cases with semantically inconsistent yet possibly high-confidence predictions that are invisible to standard attacks. As we show, standard white-box attack techniques can still easily and efficiently be applied, producing meaningful attacks and a more constrained adversarial space as compared to traditional AAs.

Contributions.   Our main contributions are:

- We introduce ConPAtt, a general XAI-supported adversarial attack goal that targets concept-based properties rather than just NN outputs.

- We proof that ConPAtt generalizes both classical targeted and untargeted AA formulations, but same-sized or smaller adversarial space.

- We hypothesize several advantages of ConPAtts for certifying robustness and for adversarial retraining, posing the chance to efficiently improve both semantic consistency and robustness.

2. Related Work

Adversarial Attacks   AAs generally search within the vicinity of an input sample x for minimally perturbed variants $\tilde{x} = x + \epsilon$ that have a malicious effect on the NN's output [17]. The perturbations can be arbitrary (digital AAs, considered) [18, 19, 20, 21, 6, 7], or further constrained to realistic changes (physical AAs) [3, 4, 22, 23, 2, 5]. However, the minimality makes the changes often invisible or difficult to see for humans. At the methodological level, black-box approaches only require access to NN inputs and outputs [23, 2, 4, 24, 25, 5]. White-box attacks as considered here instead exploit NN model internals, such as the gradient, for a more efficient search [18, 19, 21, 3, 6, 7]. Generally, AAs can be seen as a subfield of NN verification that falsifies a continuity property [26, 27]. Thus, usual search, reachability analysis, and—most prominently—optimization techniques are applicable to find or disprove adversarial examples [17]. Regarding types of specifications beyond continuity properties, approaches such as Scenic [28] and VerifAI [29] demonstrate how formal specifications can be used to generate and analyze simulation-based scenarios with symbolic inputs. In contrast, our approach targets AAs on non-symbolic image inputs, which prevents the direct use of such tools but similarly requires formal specifications.

Concept-based Explainability   Concept-based explainability generally aims to associate human-interpretable concepts with representations in NN latent space [30, 31, 32]. This includes understanding which concepts are relevant to the decision and to what extent [33, 16], and how these can be accurately recognized in NNs [14, 34]. If concept definitions in form of labeled samples are available at training time, ante-hoc approaches [35, 33, 36, 37, 38] can train individual neurons to activate for the concept. We here instead consider post-hoc approaches: These train a simple model to predict the concept of interest from an NN layer's activation [39]. Other than single-neuron-associations [40, 41], or complex models [42, 43], linear models considered here [44, 39, 45, 46] pose a good tradeoff between capturing the entanglement of representations [44, 47], interpretability [39], and favorably simple representation of the concept as halfspace in the NN's latent space.

XAI and Verification Prior work has shown that concept-based explanation methods are vulnerable to adversarial attacks. Perturbations can mislead attribution [48] and concept-based tools [49, 50], and adversarial examples significantly alter the internal concept composition of NNs [49], confirming the general fragility of interpretability methods [51]. However, these studies target concepts in isolation, without considering their joint relation to model predictions.

Beyond highlighting vulnerabilities, concept outputs have also been used for verification. Mangal et al. [52] employed vision–language models to check concept-based properties. While expressive, this approach relies on semantic similarity in multimodal embeddings (e.g., CLIP [53]), which can introduce linguistic ambiguity as well as imprecision for similar terms with small visual differences, e.g., `circle` versus `octagon`. Moreover, it is restricted to the latent space of a specific layer, although simple visual concepts may predominantly appear earlier and diminish in later layers. Cheng et al. [54] proposed specifications close to the output layer, but without decomposing them into underlying concepts and by employing an additional NN. Semantic losses [55, 12] like logic tensor networks [12] suggest to directly train concept-based rules into the network. These techniques, however, are only used for updating the NN, not for verification as done in [9], and not for AAs. Furthermore, they rely on concepts being direct outputs of the NN. Even further decoupling the verification from the NN's learned representations and thus exacerbating training efforts, Xie et al. [56] even trained completely separate NNs for predicting the concepts. Our work also directly addresses the relationship between concepts and model outputs like, a perspective that has received little attention so far [9]. However, similar to the verification testing techniques from [54, 9], we suggest to keep training and verification efforts low by using faithful explainability techniques to access concept predictions, and we newly apply the setup to AAs.

3. Background

Adversarial Attacks Let $\mathbf{x} \in \mathcal{X}$ be a real image, $\mathbf{y} \in \mathcal{Y}$ be its true label, and $f \colon \mathcal{X} \to \mathcal{Y}$ be a NN. An AA seeks an adversarial example $\mathbf{x}^{\mathrm{adv}} := \mathbf{x} + \epsilon \in \mathcal{X}$ so that its output is (sufficiently) different from the original, and the perturbation ϵ is minimal to an objective function o (usually the L1, L2, or L-infinity norm on the input for digital attacks). Sufficient difference can be formulated in terms of a $\mathbf{y}$-specific partition of the output set $\mathcal{Y}$ into a benign output set $\mathcal{Y}^+ \subset \mathcal{Y}$ with $f(\mathbf{x}) \in \mathcal{Y}^+$, and a malicious one $\mathcal{Y}^- := \mathcal{Y} \setminus \mathcal{Y}^+$. The search for the minimum perturbation ϵ then is the optimization problem

$$\mathrm{argmin}_\epsilon \, o(\epsilon) \qquad \text{s.t. } f(\mathbf{x} + \epsilon) \in \mathcal{Y}^- \,. \tag{1}$$

Adversarial attack strategies for classification are categorized as *targeted* or *untargeted* according to their choice of $\mathcal{Y}^-$: Let $p_l \colon \mathcal{Y} \to [0,1]$ denote the confidence assigned to class l, and $\theta_l \in [0,1]$ the threshold required to accept class l. In untargeted attacks, the goal is to reduce the confidence of the true class below threshold, i.e., $\mathcal{Y}^- = \{\mathbf{y} \in \mathcal{Y} \mid p_l(\mathbf{y}) < \theta_l\}$. In contrast, targeted attacks aim to raise the confidence of an incorrect class l' above a threshold, i.e., $\mathcal{Y}^- = \{\mathbf{y} \in \mathcal{Y} \mid p_{l'}(\mathbf{y}) \geq \theta_{l'}\}$.

Post-hoc Concept Extraction Let C be a set of concepts (e.g., $C = \{\mathtt{red}, \mathtt{orthogonal}\}$), and assume a possibly small classification dataset $\mathcal{D}_c = ((\mathbf{x}_k, \mathbf{y}_{c,k}))_k$ is available per concept $c \in C$. Further denote by $f_{i \to j} \colon \mathcal{X}_i \to \mathcal{X}_j$ the NN part that maps from the ith to the jth layer. Through linear post-hoc concept extraction, additional concept outputs are added to the NN by attaching for each c a linear classification model $f_{i \to c} \colon \mathcal{X}_i \to \mathcal{C}_c = [0,1]$ to the ith hidden layer as illustrated in Figure 1. Keeping the NN's weights fixed, the weights of $f_{i \to c}$ are trained on pairs $((f_{\to i}(\mathbf{x}_k), \mathbf{y}_{c,k}))_k$, such that c's concept function $f_c = f_{i \to c} \circ f_{\to i} \colon \mathcal{X} \to \mathcal{C}_c$ correctly predicts presence of the concept in an input image. Note that $f_{i \to c}$ being linear conveniently makes any subspace $\{v \in \mathcal{X}_i \mid f_{i \to c}(v) > \theta\}$ an affine linear half-space. In the following, we denote by $f_C = (f_c)_{c \in C} \colon \mathcal{X} \to \mathcal{C} = (\mathcal{C}_c)_{c \in C}$ the complete prediction of all concepts, and by $\mathcal{Z} = \mathcal{Y} \times \mathcal{C}$ the complete output set after attaching the concept outputs.

T-Norm Fuzzy Logic The standard Boolean logical connectives (*and* $\wedge$, *or* $\vee$, *not* $\neg$) can only operate on binary truth values in $\mathbb{B} = \{0, 1\}$. T-norm fuzzy logics extend the connectives to many-valued
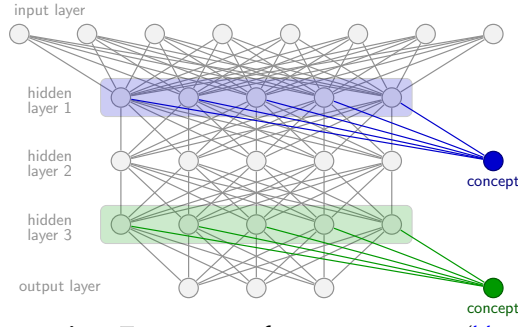
Figure 1: Post-hoc Concept Extraction: Two neurons for concept output (blue and green neurons) are post-hoc added to the trained NN (gray) using newly trained connections to hidden layers 1/3.

truth values in $\mathbb{B} = [0, 1]$ using a so-called t-norm $\wedge_t\colon [0, 1] \times [0, 1] \to [0, 1]$ to replace the $\wedge$. A valid t-norm must be monotonic, commutative and associative, have a neutral element (the 1), and match $\wedge$ on Boolean values. Typical choices for $a \wedge_t b$ are Product ($a \cdot b$), Łukasiewicz ($\max(0, a + b - 1)$), and Gödel ($\min(a, b)$) t-norms [57], since these form a generating system for all continuous t-norms. Given a $\wedge_t$, then $\neg a \coloneqq 1 - a$, $\vee_t$, $\implies_t\colon [0, 1]^2 \to [0, 1]$ can be derived and maintain desirable properties, giving the resulting t-norm logic.

Desirable properties for use of t-norm logic with NN classification outputs are: (1) The NN typically produces a confidence prediction in $[0, 1]$ instead of a Boolean value, which can be propagated by t-norm fuzzy logic to the confidence of entire logical expressions. (2) The classicale piece-wise continuous t-norm logic connectives are also piece-wise differentiable like ReLU activations of NNs. So, they can directly be used in backpropagation [12].

4. Approach

In this chapter we first define our new notion of concept-based AAs. Then we show that standard AAs are a special case, and existing attack techniques can easily be adopted to our new attack.

4.1. Concept-based Property Attacks

These classical AA types can also be interpreted as special cases of *property attacks*, where class predictions are treated as logical literals. Using fuzzy logic (see paragraph 3), we can evaluate logical expressions over outputs using a function $\text{solve}_\rho\colon \mathcal{Y} \to \mathbb{B}$ that returns the truth value of a property ρ. A property attack falsifies a given property, i.e., $\mathcal{Y}^- = \mathcal{Y}_\rho^- = \{\mathbf{y} \in \mathcal{Y} \mid \text{solve}_{\neg\rho}(\mathbf{y})\}$. Untargeted and targeted attacks correspond to properties $\rho = l$ and $\rho = \neg l$, respectively.

This perspective allows adversarial examples to be crafted with higher-order conditions — e.g., enforcing both "dog" (d) and "cat" (c) simultaneously. The corresponding attacked property is its logical negation: $\rho = \neg d \vee \neg c$.

The point of view of property attacks can also be applied to NNs that are boosted with XAI techniques. The additional concept outputs can also be used as well as the original task output of the NN to define property attacks— *Concept-based Property Attacks* (ConPAtt). For denoting the properties we propose to use the following intuitive and convenient implication format generalizing our introductory examples (all logical expressions can be reformulated like this, see Lemma 1). Note that for simplicity we shorten $f_c(\mathbf{x})$ to c, and $(\neg)c$ shorthands possibly negated c.

Lemma 1. *Each logical expression φ with two disjoint literal sets C and L can be reformulated into a term of conjunctively linked implication terms where antecedents consist only of conjunctively linked, possibly negated literals of C, and consequences consist only of disjunctively linked, possibly negated literals of L.*

Proof. Each logical expression can be reformulated into the conjunctive normal form $\varphi \equiv \bigwedge_{i>0}(\bigvee_{c \in C_i \subseteq C}(\neg)c \bigvee_{l \in L_i \subseteq L}(\neg)l)$. Let us introduce two additional variable families α_i, β_i that con-

dense the disjunctive subformulas:

$$\alpha_i := \neg \bigvee_{c \in C_i \subseteq C} (\neg)c \equiv \bigwedge_{c \in C_i \subseteq C} (\neg)c \qquad\qquad \beta_i := \bigvee_{l \in L_i \subseteq L} (\neg)l \qquad\qquad (2)$$

The subformulas can be replaced by these variables and the whole logical expression φ reformulates to $\varphi \equiv \bigwedge_{i>0} (\neg\alpha_i \vee \beta_i) \equiv \bigwedge_{i>0} (\alpha_i \implies \beta_i)$. $\qquad\square$

Definition 1 (Concept-based property). A concept-based property φ is a logical expression with two disjoint literal sets C—the concept literals—and L—the task literals—in the form of conjunctively linked implication terms whose antecedents consist only of conjunctively linked, possibly negated concept literals and whose consequences consist only of disjunctively linked, possibly negated task literals.

$$\varphi := \bigwedge_{i>0} (\alpha_i \implies \beta_i) \ , \quad \text{with} \quad \alpha_i := \bigwedge_{c \in C_i \subseteq C}(\neg)c \quad \text{and} \quad \beta_i := \bigvee_{l \in L_i \subseteq L}(\neg)l \qquad (3)$$

Definition 2 (Concept-based Property Attack). Let $\text{solve}_\varphi : \mathcal{Z} \to \mathbb{B}$ be the function to calculate the truth value of a concept-based property φ which evaluates to true at an input $\mathbf{x}$, and o a minimality measure for perturbations ϵ. A Concept-based Property Attack of φ is the search for a o-minimal perturbation ϵ to an input $\mathbf{x}$ into an adversarial example $\mathbf{x}^{\text{adv}} = \mathbf{x} + \epsilon$ which falsifies φ, i.e., lies in the malicious output set

$$\mathcal{Z}^- = \mathcal{Z}_\varphi^- = \{\mathbf{z} \in \mathcal{Z} \mid \text{solve}_{\neg\varphi}(\mathbf{z})\} \qquad (4)$$

Intuitively, a ConPAtt adversarial example $\mathbf{x}^{\text{adv}}$ to $\phi = (\bigwedge_c c \implies \bigvee_l l)$ like $\texttt{red} \wedge \texttt{octogonal} \implies \texttt{stop\_sign}$, causes the NN to predict all c as true, and all l as false. This can happen if (1) some c is predicted true even though it should be false (e.g., $\texttt{red}$ predicted true even though the change ϵ turned the sign $\texttt{gray}$), and/or (2) some l is predicted negative even though it should be positive (e.g., $\texttt{stop\_sign}$ flipped to false).

4.2. ConPAtts as Generalized Adversarial Attacks

Note that falsifying one implication term is enough to falsify a concept-based property and thus, it is sufficient to consider one implication $\phi = \alpha \implies \beta$ for an attack. The set of adversarial example task outputs can be derived from this definition, i.e. $\mathcal{Y}_\varphi^- := \{\mathbf{y} \in \mathcal{Y} \mid (\mathbf{y}, \mathbf{c}) \in \mathcal{Z}_\varphi^-\}$. Furthermore:

Theorem 1. *Standard targeted and untargeted AAs are special cases of ConPAtt.*

Proof. First note the two special cases of ConPAtt where only a single task literal is used:
1. **Generalized untargeted AAs:** $\alpha \implies l$.
2. **Generalized targeted AAs:** $\alpha \implies \neg l$.

Un-/targeted respective are generalized un-/targeted AAs with $\alpha \equiv \text{true}$, i.e., no concept restriction. $\quad\square$

A neat property of ConPAtts is that the search space is generally reduced compared to vanilla AAs:

Theorem 2. *The task output spaces of adversarial examples for generalized untargeted/targeted AAs are smaller than or equal to those for standard untargeted/targeted AAs.*

$$\mathcal{Y}_{\alpha \implies l}^- \subseteq \mathcal{Y}_l^- \qquad \mathcal{Y}_{\alpha \implies \neg l}^- \subseteq \mathcal{Y}_{\neg l}^-$$

Proof. Let us first look at generalized untargeted AA properties like $\alpha \implies l$. Each adversarial example must lack class prediction l but requires concept predictions α, i.e., they satisfy the property $\alpha \wedge \neg l$. In contrast to that, standard untargeted AAs only require the misclassification of l, i.e. each adversarial example satisfies $\neg l$ and they accept adversarial examples that do not additionally fulfill α. It follows that the valid output space of adversarial examples for generalized untargeted AAs $\mathcal{Z}_{\alpha \implies l}^-$ is smaller than or equal to that for standard untargeted AAs $\mathcal{Z}_l^-$ as well as for their valid task output spaces $\mathcal{Y}_{\alpha \implies l}^- \subseteq \mathcal{Y}_l^-$.

In this explanation, it does not matter whether both adversarial examples expect a misclassification $\neg l$ or a specific task output l. That is why this relation also applies between generalized targeted AAs and standard targeted AAs, i.e. $\mathcal{Y}_{\alpha \implies \neg l}^- \subseteq \mathcal{Y}_{\neg l}^-$. $\quad\square$

ConPAtt Procedure   ConPAtt can be easily performed with any existing AA approach. The trick is to use the result of the (partially) differentiable fuzzy operation $\varphi \circ (f, f_C) \colon \mathcal{X} \to \mathbb{B}$ instead of the output of the NN. This makes ConPAtt a targeted AA with the expected result False or 0 for the adversarial examples.

5. Discussion and Outlook: ConPAtt for Adversarial Training

In the following we discuss further what practical benefits we expect from this more general formulation of attack goals, how this could be evaluated, and which challenges are still open.

5.1. Hypothesized Benefits of ConPAtts

We hypothesize that
- *generalized (un-)targeted AAs with at least one concept **reduce the search space** for adversarial examples not only theoretically but also empirically,*
- *the adversarial examples obtained via ConPAtt are **particularly efficient for retraining** because they are pinpoint adversarial examples with a high information content.*

ConPAtts versus Standard AAs: To understand above claims, one should first have a closer look at the vulnerabilities that can be exploited for a successfull ConPAtt attack against a concept-based property $\phi = (\alpha \implies l)$. Standard AAs capture any cases, where the final output l is changed, regardless of whether this resulted in illogical behavior breaking ϕ or not. Thus, standard AAs may primarily focus on turning off causally related early-layer concepts, i.e., falsifying α to falsify l. For example, falsify `red` to cause a negative output of `stop_sign`. This is not sufficient for a ConPAtt to ϕ, for which not only l must become false, but simultaneously α must remain true (cf. Theorem 2). It is therefore not guaranteed that one obtains the same results for ConPAtts against any of the following concept-based properties:
- $\phi_l = (\text{true} \implies l)$, which is the standard AA against the output l,
- $\phi_\alpha = (\neg\alpha \implies \text{false})$, which is the standard AA against the concept outputs, i.e., the attack flips any concept c in the conjunction $\alpha = \bigwedge_c c$ to false, and
- $\phi = (\alpha \implies l)$, which is a generalized concept-based property attack.

Whether the obtained adversarial examples are similar depends on whether it is easier to attack concepts—then falsifying ϕ_l and ϕ_α should yield similar results—or logics, in which case falsifying ϕ_l and ϕ are expected to yield similar results. Since concepts themselves represent noisy variables with non-perfect accuracy, chances are high that attacking concepts generally is easier than attacking logics. Our ConPAtt framework provides the option to test and train on these different rules individually, and hence distinguish more finegrained between simply attacking the concepts or outputs, and truly attacking internal logics.

Benefits of Targeting Logics: One reason for both of the claims is on semantic level: Human-defined properties typically encode important knowledge about the task at hand, thus should strengthen both the adherence to the properties and indirectly the actual main task of the network. Given that well-generalizing NNs typically adopt this knowledge to large extend, the cases of logic breaches should be few but meaningful. This would make **attacking logics especially beneficial for retraining purposes** similar to adversarial training [17, 58].

Benefits for Computational Efficiency: Also, here directly benefit from low integration overhead: (1) Preparation only requires cheap post-hoc concept extraction; (2) Only very few additional operations (the $f_{i \to c}$) are added that need backpropagation/-tracing if gradient-based attack methods are used; and (3) The beneficial formulation of concepts as half-spaces in latent spaces allows efficient reachability analysis with substantial reduction in the search space as illustrated in Figure 2 and sketched in
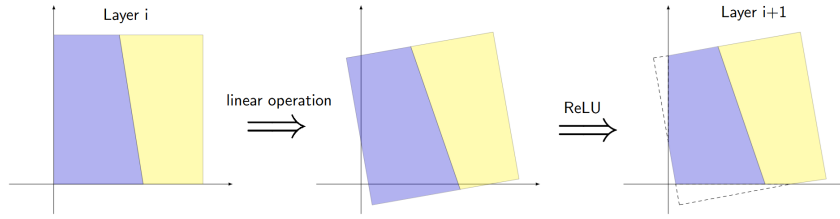
Figure 2: Concept Propagation: Illustration how the concept and non-concept half-spaces propagate from layer to layer using linear operations like convolutions (left to mid) combined with ReLU activation (mid to right). Concretely, ReLUs add additional bends of wide angle to the decision boundary.

Appendix A. Next steps should empirically test the attack success and the effect of retraining with adversarial examples of this approach.

5.2. Future Work: Evaluation and Challenges

Planned Experimental Setting: We suggest to evaluate several aspects to ensure a comprehensive assessment. As metrics, we consider both task performance and rule adherence, measured through accuracy and Intersection-over-Union (IoU) for task prediction as well as rule satisfaction. In addition, we track the success of adversarial attacks before retraining, as well as the effectiveness of defences and the accuracy of concepts after retraining. For evaluation, we draw on three established **datasets**: MNIST [59], GTSRB [60], and ImageNet [61]. The **models** include self-trained simple architectures for MNIST and GTSRB, as well as a range of widely used ImageNet classifiers: Inception-v3 [62], Inception-v4 [63], Inception-Resnet-v2 [63], Resnet-v2-101 [64] and the ensemble-based variants Inception $v3_{ens3}$, Inception $v3_{ens4}$ and IncRes $v2_{ens}$ [58]. For baselines, we rely on several state-of-the-art adversarial attack methods, namely SGM [19], VMI-FGSM and VNI-FGSM [21], L2T [6], and BSR [7].

The attacked concept-based properties reflect both simple and more complex relations. Examples include that class 1 implies the concept `line`, that classes 1 and 2 should never be predicted simultaneously (i.e., $\neg 1 \lor \neg 2$), and that the concepts `red`, `octagon`, and `stop_label` together imply `stop_sign`.

Challenges and further Future Work: As explained above, it is expected that ConPAtts not necessarily yield the same results as standard AAs that attack outputs or concepts. In addition to above experiments, one could contrastively compare results for the different attacks for insights how large the gap truly is. However, a considerable challenge for the experimental evaluation is that retraining procedures may need to be adapted: (Adversarially) retraining with respect to the *task output* might accidentally destroy the post-hoc attached *concept outputs*. Countermeasures might be to freeze earlier NN parts up to the concept prediction, or alternatingly or simultaneously retrain the NN and the concept predictors. Experiments must show how to balance need for concept labels with concept accuracy during adversarial finetuning.

6. Conclusion

In this position paper, we introduce a novel generalized adversarial attack goal: Instead of targeting a change in (respectively falsification of) the output class, our attacks aim to falsify the compliance of the NN with prior symbolic knowledge on sufficient indicators for an output class. Standard AAs are shown to be a specific case of our generalized formulation for concept-based properties. Also, these allow to substantially reduce the expected search space of the AA search with increasing number of concepts. Also, we argue that these concept-based properties provide a more natural and human-aligned target for AAs. This suggests that they might be particularly suited for NN robustification via adversarial model (re)training or runtime monitoring.

Acknowledgments

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT based on GPT-4o in order to: Improve writing style. After using these tool(s)/service(s), the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

A. Considerations for Reachability-based Search

Existing reachability-based techniques conduct forward and/or backward passes through the NN to trace / estimate regions of interest through the NN processing. We here show how the considered concept-based properties give rise to a particularly efficient formulation of this approach: Being half-spaces in intermediate layers, the (negated) concepts have the potential to easily and substantially reduce the adversarial space that one needs to keep track off half-way through the network and can also be easily described in later layers as sketched in Figure 2. In the following, this is illustrated for a back-propagation approach for a simple generalized untargeted attack $(c_1 \wedge \cdots \wedge c_n) \implies l$. Recall that a valid counterexample falsifying the property $(c_1 \wedge \cdots \wedge c_n) \implies l$ must fulfil $c_1 \wedge \cdots \wedge c_n \wedge \neg l$.

Denote by $f_{\mathcal{L} \to \mathcal{L}'} \colon \mathcal{L} \to \mathcal{L}'$ the NN part mapping from layer $\mathcal{L}$ to $\mathcal{L}'$, and $p_{\mathcal{L}' \to c}^{(\mathcal{L})} = p_c \circ f_{\mathcal{L} \to \mathcal{C}} \circ f_{\mathcal{L}' \to \mathcal{L}} \colon \mathcal{L}' \to [0,1]$ the function evaluating the presence of concept c in layer $\mathcal{L}$ for a latent vector v from an earlier layer $\mathcal{L}'$. Denote by $\mathcal{L}_c$ the layer which was chosen for the embedding of concept c, and let $\mathcal{L}_1$ be the earliest layer for which $\mathcal{L}_1 = \mathcal{L}_c$ for some c. Note that $\mathcal{L}_l = \mathcal{L}_{L-1}$ is the final representation layer before the output confidence prediction, if this is L layers later than $\mathcal{L}_1$. Let $H_c = \{v \in \mathcal{L}_c \mid p_{\mathcal{L}_c \to c}(v) < \theta_c\}$ be the halfspace of the concept c in the concept's $\mathcal{L}_c$.

Now we can reformulate the falsification as a search for a region in latent space:

Lemma 2. *A representation $v = f_{\to \mathcal{L}}(x) \in \mathcal{L}$ in layer $\mathcal{L}$ of a valid counterexample $x \in \mathcal{X}$ to the concept-based property $(c_1 \wedge \cdots \wedge c_n) \implies l$ must fulfil $v \in \bigwedge_{c \in c_i, l} f_{\mathcal{L} \to \mathcal{L}_c}^{-1}(H_c)$.*

While it is costly to determine $f_{\mathcal{L} \to \mathcal{L}_c}^{-1}(H_c)$ independently, the concept-based property gives rise to a recursive definition:

Theorem 3. *Recursively define the propagation of halfspace intersections through the NN*

$$P_{L-1} = \bigcap_{\mathcal{L}_L = \mathcal{L}_c} H_c \,, \qquad\qquad P_i = f_{\mathcal{L}_{i-1} \to \mathcal{L}_i}^{-1}(P_{i+1}) \cap \bigcap_{\mathcal{L}_i = \mathcal{L}_c} H_c \qquad (5)$$

Then for any counterexample x to above concept-based property it must hold that $f_{\to \mathcal{L}}(x) \in P_1$. P_1 can be efficiently calculated using a single backward propagation through layers $L - 1$ to 1.

Proof. The property inductively follow from the definition, noting that $P_1 = \bigwedge_{c \in c_i, l} f_{\mathcal{L} \to \mathcal{L}_c}^{-1}(H_c)$ and the constraint of considering ReLU networks. $\qquad \square$

In particular, each propagation step only requires to obtain a polytope's preimage for a single NN layer operation, and apply a cheap intersection of the resulting polytope with halfspaces. This makes the first part of the search very efficient, promising speedup compared to a full end-to-end search for counterexamples directly in the input space.

The forward-propagation case is similar. Here, it can additionally be shown, that the propagated P_i always is a connected polytope, since intersection with half-spaces does not change this property, neither does the forward pass through continuous layer operations.

References

[1] P. Rech, Artificial Neural Networks for Space and Safety-Critical Applications: Reliability Issues and Potential Solutions, IEEE Transactions on Nuclear Science 71 (2024) 377–404. URL: https://ieeexplore.ieee.org/abstract/document/10380628. doi:10.1109/TNS.2024.3349956.

[2] N. Suryanto, Y. Kim, H. Kang, H. T. Larasati, Y. Yun, T.-T.-H. Le, H. Yang, S.-Y. Oh, H. Kim, DTA: Physical Camouflage Attacks Using Differentiable Transformation Network, 2022, pp. 15305–15314. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Suryanto_DTA_Physical_Camouflage_Attacks_Using_Differentiable_Transformation_Network_CVPR_2022_paper.html.

[3] Y. Li, Y. Li, X. Dai, S. Guo, B. Xiao, Physical-World Optical Adversarial Attacks on 3D Face Recognition, 2023, pp. 24699–24708. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Li_Physical-World_Optical_Adversarial_Attacks_on_3D_Face_Recognition_CVPR_2023_paper.html.

[4] C. Hu, Y. Wang, K. Tiliwalidi, W. Li, Adversarial Laser Spot: Robust and Covert Physical-World Attack to DNNs, in: Proceedings of The 14th Asian Conference on Machine Learning, PMLR, 2023, pp. 483–498. URL: https://proceedings.mlr.press/v189/hu23b.html, iSSN: 2640-3498.

[5] J. Zheng, C. Lin, J. Sun, Z. Zhao, Q. Li, C. Shen, Physical 3D Adversarial Attacks against Monocular Depth Estimation in Autonomous Driving, 2024, pp. 24452–24461. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Zheng_Physical_3D_Adversarial_Attacks_against_Monocular_Depth_Estimation_in_Autonomous_CVPR_2024_paper.html.

[6] R. Zhu, Z. Zhang, Z. Liu, C. Xu, S. Liang, Learning to Transform Dynamically for Better Adversarial Transferability, 2024. URL: https://openreview.net/forum?id=k76ngWX9OR.

[7] K. Wang, X. He, W. Wang, X. Wang, Boosting Adversarial Transferability by Block Shuffle and Rotation, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 24336–24346. URL: https://ieeexplore.ieee.org/abstract/document/10656871. doi:10.1109/CVPR52733.2024.02297, iSSN: 2575-7075.

[8] D. Ming, P. Ren, Y. Wang, X. Feng, Boosting the Transferability of Adversarial Attack on Vision Transformer with Adaptive Token Tuning, 2024. URL: https://openreview.net/forum?id=sNz7tptCH6.

[9] G. Schwalbe, C. Wirth, U. Schmid, Enabling verification of deep neural networks in perception tasks using fuzzy logic and concept embeddings, 2022. doi:10.48550/arXiv.2201.00572. arXiv:2201.00572.

[10] E. Giunchiglia, M. Stoian, S. Khan, F. Cuzzolin, T. Lukasiewicz, ROAD-R: The Autonomous Driving Dataset with Logical Requirements, in: IJCLR 2022 Workshops, Vienna, Austria, 2022.

[11] A. Ledaguenel, C. Hudelot, M. Khouadjia, Improving Neural-based Classification with Logical Background Knowledge, in: ECAI 2024 Workshop Proceedings, arXiv, Santiago de Compostela, Spain, 2024. arXiv:2402.13019.

[12] S. Badreddine, A. d'Avila Garcez, L. Serafini, M. Spranger, Logic Tensor Networks, Artificial Intelligence 303 (2022) 103649. doi:10.1016/j.artint.2021.103649.

[13] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network Dissection: Quantifying Interpretability of Deep Visual Representations, 2017, pp. 6541–6549. URL: https://openaccess.thecvf.com/content_cvpr_2017/html/Bau_Network_Dissection_Quantifying_CVPR_2017_paper.html.

[14] R. Fong, A. Vedaldi, Net2Vec: Quantifying and Explaining How Concepts Are Encoded by Filters in Deep Neural Networks, 2018, pp. 8730–8738. URL: https://openaccess.thecvf.com/content_cvpr_2018/html/Fong_Net2Vec_Quantifying_and_CVPR_2018_paper.html.

[15] J. Crabbé, M. van der Schaar, Concept Activation Regions: A Generalized Framework For Concept-Based Explanations, Advances in Neural Information Processing Systems 35 (2022) 2590–2607. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/11a7f429d75f9f8c6e9c630aeb6524b5-Abstract-Conference.html.

[16] T. Oikarinen, T.-W. Weng, CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks, 2022. URL: https://openreview.net/forum?id=iPWiwWHc1V.

[17] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, H. W. Alomari, Adversarial Deep Learning:

A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification, IEEE Access 10 (2022) 102266–102291. doi:`10.1109/ACCESS.2022.3208131`.

[18] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2014. URL: http://arxiv.org/abs/1312.6199. doi:`10.48550/arXiv.1312.6199`, arXiv:1312.6199 [cs].

[19] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, X. Ma, Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets, 2019. URL: https://openreview.net/forum?id=BJlRs34Fvr.

[20] J. Su, D. V. Vargas, K. Sakurai, One Pixel Attack for Fooling Deep Neural Networks, IEEE Transactions on Evolutionary Computation 23 (2019) 828–841. URL: https://ieeexplore.ieee.org/document/8601309. doi:`10.1109/TEVC.2019.2890858`, conference Name: IEEE Transactions on Evolutionary Computation.

[21] X. Wang, K. He, Enhancing the Transferability of Adversarial Attacks Through Variance Tuning, 2021, pp. 1924–1933. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Enhancing_the_Transferability_of_Adversarial_Attacks_Through_Variance_Tuning_CVPR_2021_paper.html.

[22] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust Physical-World Attacks on Deep Learning Visual Classification, 2018, pp. 1625–1634. URL: https://openaccess.thecvf.com/content_cvpr_2018/html/Eykholt_Robust_Physical-World_Attacks_CVPR_2018_paper.

[23] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, D. Tao, Perceptual-Sensitive GAN for Generating Adversarial Patches, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 1028–1035. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3893. doi:`10.1609/aaai.v33i01.33011028`, number: 01.

[24] W. Huang, X. Zhao, G. Jin, X. Huang, SAFARI: Versatile and Efficient Evaluations for Robustness of Interpretability, 2023, pp. 1988–1998. URL: https://openaccess.thecvf.com/content/ICCV2023/html/Huang_SAFARI_Versatile_and_Efficient_Evaluations_for_Robustness_of_Interpretability_ICCV_2023_paper.html.

[25] D. Wang, W. Yao, T. Jiang, C. Li, X. Chen, RFLA: A Stealthy Reflected Light Adversarial Attack in the Physical World, 2023, pp. 4455–4465. URL: https://openaccess.thecvf.com/content/ICCV2023/html/Wang_RFLA_A_Stealthy_Reflected_Light_Adversarial_Attack_in_the_Physical_ICCV_2023_paper.html.

[26] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, Algorithms for verifying deep neural networks, Foundations and Trends® in Optimization 4 (2021) 244–404. doi:`10.1561/2400000035`. arXiv:`1903.06758`.

[27] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, D. Kroening, Concolic testing for deep neural networks, in: Proc. 33rd ACM/IEEE Int. Conf. Automated Software Engineering, ACM, Montpellier, France, 2018, pp. 109–119. doi:`10.1145/3238147.3238172`.

[28] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, S. A. Seshia, Scenic: a language for scenario specification and scene generation, in: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2019, Association for Computing Machinery, New York, NY, USA, 2019, pp. 63–78. URL: https://dl.acm.org/doi/10.1145/3314221.3314633. doi:`10.1145/3314221.3314633`.

[29] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, S. A. Seshia, VerifAI: A Toolkit for the Formal Design and Analysis of Artificial Intelligence-Based Systems, in: I. Dillig, S. Tasiran (Eds.), Computer Aided Verification, Springer International Publishing, Cham, 2019, pp. 432–442. doi:`10.1007/978-3-030-25540-4_25`.

[30] J. H. Lee, G. Mikriukov, G. Schwalbe, S. Wermter, D. Wolter, Concept-Based Explanations in Computer Vision: Where Are We and Where Could We Go?, in: A. Del Bue, C. Canton, J. Pont-Tuset, T. Tommasi (Eds.), Computer Vision – ECCV 2024 Workshops, Springer Nature Switzerland, Cham, 2025, pp. 266–287. doi:`10.1007/978-3-031-92648-8_17`.

[31] E. Poeta, G. Ciravegna, E. Pastor, T. Cerquitelli, E. Baralis, Concept-based Explainable Artificial

Intelligence: A Survey, 2023. doi:10.48550/arXiv.2312.12936. arXiv:2312.12936.

[32] G. Schwalbe, Concept Embedding Analysis: A Review, 2022. doi:10.48550/arXiv.2203.13909. arXiv:2203.13909.

[33] A. Wan, L. Dunlap, D. Ho, J. Yin, S. Lee, S. Petryk, S. A. Bargal, J. E. Gonzalez, NBDT: Neural-Backed Decision Tree, 2020. URL: https://openreview.net/forum?id=mCLVeEpplNE.

[34] G. Schwalbe, Verification of Size Invariance in DNN Activations Using Concept Embeddings, in: I. Maglogiannis, J. Macintyre, L. Iliadis (Eds.), Artificial Intelligence Applications and Innovations, volume 627, Springer International Publishing, Cham, 2021, pp. 374–386. URL: https://link.springer.com/10.1007/978-3-030-79150-6_30. doi:10.1007/978-3-030-79150-6_30.

[35] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, P. Liang, Concept Bottleneck Models, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348. URL: https://proceedings.mlr.press/v119/koh20a.html.

[36] M. Yuksekgonul, M. Wang, J. Zou, Post-hoc Concept Bottleneck Models, 2022. URL: https://openreview.net/forum?id=nA5AZ8CEyow.

[37] T. Oikarinen, S. Das, L. M. Nguyen, T.-W. Weng, Label-free Concept Bottleneck Models, 2022. URL: https://openreview.net/forum?id=FlCg47MNvBA.

[38] Y. Yang, A. Panagopoulou, S. Zhou, D. Jin, C. Callison-Burch, M. Yatskar, Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification, 2023, pp. 19187–19197. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Yang_Language_in_a_Bottle_Language_Model_Guided_Concept_Bottlenecks_for_CVPR_2023_paper.html.

[39] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), in: Proc. 35th Int. Conf. Machine Learning, volume 80 of Proceedings of Machine Learning Research, PMLR, Stockholmsmässan, Stockholm, Sweden, 2018, pp. 2668–2677.

[40] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: Proc. 2017 IEEE Conf. Comput. Vision and Pattern Recognition, IEEE Computer Society, Honolulu, HI, USA, 2017, pp. 3319–3327. doi:10.1109/CVPR.2017.354. arXiv:1704.05796.

[41] C. Olah, A. Mordvintsev, L. Schubert, Feature visualization, Distill 2 (2017) e7. doi:10.23915/distill.00007.

[42] J. Crabbé, M. van der Schaar, Concept Activation Regions: A Generalized Framework For Concept-Based Explanations, Advances in Neural Information Processing Systems 35 (2022) 2590–2607.

[43] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, B. I. P. Rubinstein, Invertible concept-based explanations for CNN models with non-negative concept activation vectors, in: Proc. 35th AAAI Conf. Artificial Intelligence, volume 35, AAAI Press, virtual, 2021, pp. 11682–11690.

[44] R. Fong, A. Vedaldi, Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks, in: Proc. 2018 IEEE Conf. Comput. Vision and Pattern Recognition, IEEE Computer Society, Salt Lake City, UT, USA, 2018, pp. 8730–8738. doi:10.1109/CVPR.2018.00910.

[45] M. Graziani, V. Andrearczyk, H. Müller, Regression concept vectors for bidirectional explanations in histopathology, in: D. Stoyanov, Z. Taylor, S. M. Kia, I. Oguz, M. Reyes, A. Martel, L. Maier-Hein, A. F. Marquand, E. Duchesnay, T. Löfstedt, B. Landman, M. J. Cardoso, C. A. Silva, S. Pereira, R. Meier (Eds.), Understanding and Interpreting Machine Learning in Medical Image Computing Applications, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 124–132. doi:10.1007/978-3-030-02628-8_14.

[46] G. Mikriukov, G. Schwalbe, K. Bade, Local Concept Embeddings for Analysis of Concept Distributions in Vision DNN Feature Spaces, International Journal of Computer Vision (2025). doi:10.1007/s11263-025-02446-y.

[47] M. Dreyer, E. Purelku, J. Vielhaben, W. Samek, S. Lapuschkin, PURE: Turning Polysemantic Neurons Into Pure Features by Identifying Relevant Circuits, in: CVPR2024 Workshops, XAI4CV, arXiv, Seattle Convention Center, Seattle, WA, USA, 2024. doi:10.48550/arXiv.2404.06453. arXiv:2404.06453.

[48] D. Slack, S. Hilgard, E. Jia, S. Singh, H. Lakkaraju, Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 180–186. URL: https://dl.acm.org/doi/10.1145/3375627.3375830. doi:10.1145/3375627.3375830.

[49] G. Mikriukov, G. Schwalbe, F. Motzkus, K. Bade, Unveiling the Anatomy of Adversarial Attacks: Concept-Based XAI Dissection of CNNs, in: L. Longo, S. Lapuschkin, C. Seifert (Eds.), Explainable Artificial Intelligence, Springer Nature Switzerland, Cham, 2024, pp. 92–116. doi:10.1007/978-3-031-63787-2_6.

[50] D. Brown, H. Kvinge, Making Corgis Important for Honeycomb Classification: Adversarial Attacks on Concept-Based Explainability Tools, 2023, pp. 620–627. URL: https://openaccess.thecvf.com/content/CVPR2023W/TAG-PRA/html/Brown_Making_Corgis_Important_for_Honeycomb_Classification_Adversarial_Attacks_on_Concept-Based_CVPRW_2023_paper.html.

[51] A. Ghorbani, A. Abid, J. Zou, Interpretation of Neural Networks Is Fragile, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 3681–3688. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4252. doi:10.1609/aaai.v33i01.33013681.

[52] R. Mangal, N. Narodytska, D. Gopinath, B. C. Hu, A. Roy, S. Jha, C. S. Păsăreanu, Concept-Based Analysis of Neural Networks via Vision-Language Models, in: G. Avni, M. Giacobbe, T. T. Johnson, G. Katz, A. Lukina, N. Narodytska, C. Schilling (Eds.), AI Verification, Springer Nature Switzerland, Cham, 2024, pp. 49–77. doi:10.1007/978-3-031-65112-0_3.

[53] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning Transferable Visual Models From Natural Language Supervision, in: Proceedings of the 38th International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html, iSSN: 2640-3498.

[54] C.-H. Cheng, C.-H. Huang, T. Brunner, V. Hashemi, Towards Safety Verification of Direct Perception Neural Networks, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 1640–1643. URL: https://ieeexplore.ieee.org/abstract/document/9116205. doi:10.23919/DATE48585.2020.9116205, iSSN: 1558-1101.

[55] J. Xu, Z. Zhang, T. Friedman, Y. Liang, G. Broeck, A semantic loss function for deep learning with symbolic knowledge, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 5502–5511.

[56] X. Xie, K. Kersting, D. Neider, Neuro-symbolic verification of deep neural networks, in: Thirty-First International Joint Conference on Artificial Intelligence, volume 4, 2022, pp. 3622–3628. doi:10.24963/ijcai.2022/503.

[57] P. Hájek, Metamathematics of Fuzzy Logic, Trends in Logic, Springer Netherlands, Dordrecht, 1998. URL: http://link.springer.com/10.1007/978-94-011-5300-3. doi:10.1007/978-94-011-5300-3, iSSN: 1572-6126.

[58] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble Adversarial Training: Attacks and Defenses, 2018. URL: https://openreview.net/forum?id=rkZvSe-RZ.

[59] L. Deng, The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web], IEEE Signal Processing Magazine 29 (2012) 141–142. URL: https://ieeexplore.ieee.org/document/6296535. doi:10.1109/MSP.2012.2211477.

[60] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, Neural Networks 32 (2012) 323–332. URL: https://www.sciencedirect.com/science/article/pii/S0893608012000457. doi:10.1016/j.neunet.2012.02.016.

[61] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision 115 (2015) 211–252. URL: https://doi.org/10.1007/s11263-015-0816-y. doi:10.1007/s11263-015-0816-y.

[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, 2016, pp. 2818–2826. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html.

[63] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, Proceedings of the AAAI Conference on Artificial Intelligence 31 (2017). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11231. doi:10.1609/aaai.v31i1.11231, number: 1.

[64] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2016, pp. 770–778. URL: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.