Right for the Right Concept: Revising Neuro-Symbolic Concepts by Interacting with their Explanations

Wolfgang Stammer*, Patrick Schramowski*& Kristian Kersting*†

December 15, 2020

Abstract

Most explanation methods in deep learning map importance estimates for a model's prediction back to the original input space. These "visual" explanations are often insufficient, as the model's actual concept remains elusive. Moreover, without insights into the model's semantic concept, it is difficult —if not impossible— to intervene on the model's behavior via its explanations, called Explanatory Interactive Learning. Consequently, we propose to intervene on a Neuro-Symbolic scene representation, which allows one to revise the model on the semantic level, e.g. "never focus on the color to make your decision". We compiled a novel confounded visual scene data set, the CLEVR-Hans data set, capturing complex compositions of different objects. The results of our experiments on CLEVR-Hans demonstrate that our semantic explanations, i.e. compositional explanations at a per-object level, can identify confounders that are not identifiable using "visual" explanations only. More importantly, feedback on this semantic level makes it possible to revise the model from focusing on these confounding factors.

1 Introduction

Machine learning models may show Clever-Hans like moments when solving a task by learning the "wrong" thing, *e.g.*making use of confounding factors within a data set. Unfortunately, it is not easy to find out whether, say, a deep neural network is making Clever-Hans-type mistakes because they are not reflected in the standard performance measures such as precision and recall. Instead, one looks at their explanations to see what features the network is actually using [24]. By interacting with the explanations, one may even fix Clever-Hans like moments [44, 51, 48, 45].

This Explanatory Interactive Learning (XIL), however, very much depends on the provided explanations. Most explanation methods in deep learning map importance estimates for a model's prediction back to the original input space [47, 50, 49, 46, 7]. This is somewhat reminiscent of a child who points towards something but cannot articulate why something is relevant. In other words, "visual" explanations are insufficient if a task requires a concept-level understanding of a model's decision. Without knowledge about and symbolic access to the concept level, it remains difficult—if not impossible—to fix Clever-Hans behavior.

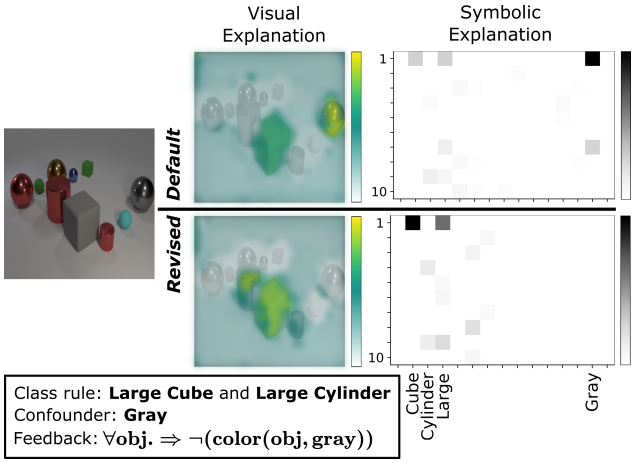

Figure 1: **Neuro-Symbolic explanations are needed to revise deep learning models from focusing on irrelevant features via global feedback rules.**

To illustrate this, consider the classification task depicted in Fig. 1. It shows a complex scene consisting of objects, which vary in position, shape, size, material, and color. Now, assume that scenes belonging to the true class show a large cube and a large cylinder. Unfortunately, during training, our deep network only sees scenes with large, gray cubes. Checking the deep model's decision process using visual explanations confirms this: the deep model has learned to largely focus on the gray cube to classify scenes to be positive. An easy fix would be to provide feedback in the form of "never focus on the color to make your decision" as it would eliminate the confounding factor. Unfortunately, visual explanations do not allow us direct access to the semantic level— they do not tell us that "the color gray is an important feature for the task at hand" and we cannot provide feedback at the symbolic level.

Triggered by this, we present the first Neuro-Symbolic XIL (NeSy XIL) approach that is based on decomposing a visual scene into an object-based, symbolic representation and, in turn, allows one to compute and interact with neuro-symbolic explanations. We demonstrate the advantages of NeSy XIL on

*Technical University of Darmstadt, Computer Science Department, Artificial Intelligence and Machine Learning Lab, Darmstadt, Germany
†Technical University of Darmstadt, Centre for Cognitive Science, Darmstadt, Germany
Contact: wolfgang.stammer@cs.tu-darmstadt.de
Preprint. Work in progress.

a newly compiled, confounded data set, called CLEVR-Hans. It consists of scenes that can be classified based on specific combinations of object attributes and relations. Importantly, CLEVR-Hans encodes confounders in a way so that the confounding factors are not separable in the original input space, in contrast to many previous confounded computer vision data sets.

To sum up, this work makes the following contributions:

- We confirm empirically on our newly compiled confounded benchmark data set, CLEVR-Hans, that Neuro-Symbolic concept learners [32] may show Clever-Hans moments, too.
- To this end, we devise a novel Neuro-Symbolic concept learner, combining Slot Attention [29] and Set Transformer [25] in an end-to-end differentiable fashion.
- Given symbolic annotations about incorrect explanations, even across a set of several instances, we efficiently optimize the Neuro-Symbolic concept learner to be right for better Neuro-Symbolic reasons.

These contributions are important to make progress towards creating conversational explanations between machines and human users [54, 34]. This is necessary for improved trust development and truly Explanatory Interactive Learning: symbolic abstractions help us, humans, to engage in conversations with one another and to convey our thoughts efficiently, without the need to specify much detail.

We proceed as follows. We start off by briefly reviewing related work. Then we introduce NeSy XIL including our novel Neuro-Symbolic concept learner by combining Slot Attention with Set Transformer. Before concluding, we touch upon the results of our experimental evaluation on our CLEVR-Hans data set.

2 Related Work on XIL

Our work touches upon Explainable AI, Explanatory Interactive Learning, and Neuro-Symbolic architectures.

Explainable AI (XAI) methods, in general, are used to evaluate the reasons for a (black-box) model's decision by presenting the model's explanation in a hopefully human-understandable way. Current methods can be divided into various categories based on characteristics [56], *e.g.*their level of intrinsicality or if they are based on back-propagation computations. Across the spectrum of XAI approaches, from backpropagation-based [50, 2], to model distillation [42], or prototype-based [26] methods, very often an explanation is created by highlighting or otherwise relating direct input elements to the model's prediction, thus visualizing an explanation at the level of the original input space.

Several studies have investigated methods that produce explanations other than these visual explanations, such as multimodal explanations [37, 55, 41], including visual and logic rule explanations [1, 40]. [33, 28] investigate methods for creating more interactive explanations, whereas [3] focuses on creating single-modal, logic-based explanations. Some recent work has also focused on creating concept-based explanations

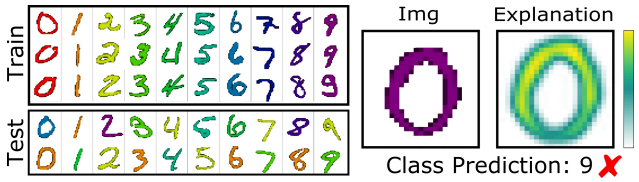

Figure 2: **Visual Explanations for ColorMNIST.** (Left) the general data distribution between train and test split. (Right) a typical visual explanation of a CNN. Notice digit pixels are considered as important for the wrong prediction.

[18, 61, 9]. None of the above studies, however, investigate using the explanations as a means of intervening on the model.

Explanatory interactive learning (XIL) [44, 48, 51, 45] merges XAI with an active learning setting. It incorporates XAI in the learning process by involving the human-user —interacting on the explanations— in the training loop. More precisely, the human user can query the model for explanations of individual predictions and respond by correcting the model if necessary, providing a slightly improved —but not necessarily optimal— feedback on the explanations. Thus, as in active learning, the user can provide the correct label if the prediction is wrong. In addition, XIL also allows the user to provide feedback on the explanation. This combination of receiving explanations and user interaction is a strong necessity for gaining trust in the model's behavior [51, 45]. XIL can be applied to differentiable as well as non-differentiable models [45].

Neuro-Symbolic architectures [8, 58, 32, 13, 53, 6] make use of data-driven, sub-symbolic representations, and symbol-based reasoning systems. This field of research has received increasing interest in recent years as a means of solving issues of individual subsystems, such as the out-of-distribution generalization problem of many neural networks, by combining the advantages of symbolic and sub-symbolic models. Yi *et al.* [58], for example, propose a Neuro-Symbolic based VQA system based on disentangling visual perception from linguistic reasoning. Each sub-module of their system processes different subtasks, *e.g.*their scene parser decomposes a visual scene into an object-based scene representation. Their reasoning engine then uses this decomposed scene representation rather than directly computing in the original input space. An approach that also relates to the work of Lampert *et al.* [22, 23].

3 Motivating Example: Color-MNIST

To illustrate the problem setting, we first revert to a well known confounded toy data set. ColorMNIST [17, 43] consists of colored MNIST digits. Within the training set, each number is confounded with a specific color, whereas in the test set, the color association is shuffled or inverted.

A simple CNN model can reach 100% accuracy on the training set, but only 23% on the test set, indicating that the model has learned to largely focus on the color for accurate prediction rather than the digits themselves. Fig. 2 depicts the visual

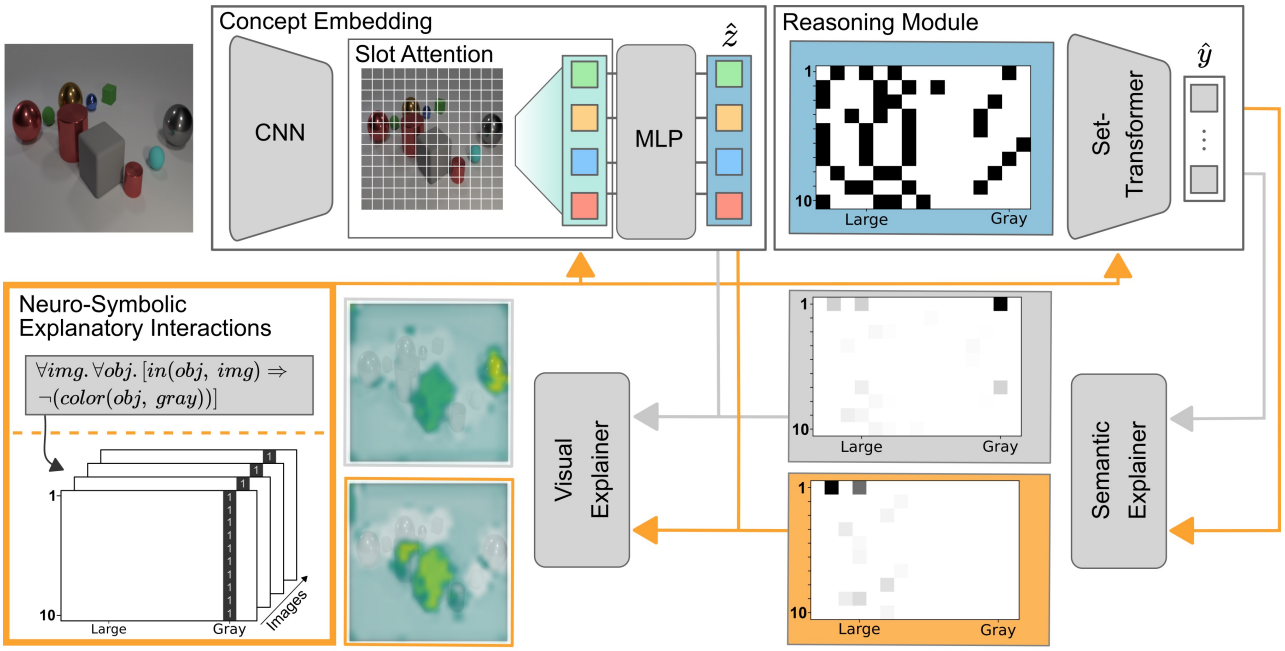

Figure 3: **Neuro-Symbolic XIL for improved explanations and interaction.** (Top) Neuro-Symbolic Concept Learner with Slot-Attention and Set Transformer. (Bottom) Neuro-Symbolic revision pipeline with explanations of the model before (gray) and after applying the feedback (orange).

explanation (here created using GradCAM [47]) of a zero that is predicted as a nine. Note the zero is colored in the same color as all nines of the training set. From the visual explanation it becomes clear that the model is focusing on the correct object, however why the model is predicting the wrong digit label does not become clear without an understanding of the underlying training data distribution.

Importantly, although the model is wrong for the right reason, it is a non-trivial problem of interacting with the model to revise its decision using XIL solely based on these explanations. Setting a loss term to correct the explanation (*e.g.*[44]) on color channels is as non-trivial and inconvenient as unconfounding the data set with counterexamples [51]. Kim *et al.* [17] describe how to unbias such a data set if the bias is known, using the mutual information between networks trained on separate features of the data set in order for the main network not to focus on bias features. Rieger *el al.* [43] propose an explanation penalization loss similar to [44, 48, 45], focusing on Contextual Decomposition [36] as explanation method. However, the utilized penalization method is task-specific and detached from the model's explanations, resulting in only a slight improvement of a final 31% accuracy (using the inverted ColorMNIST setting).

4 Neuro-Symbolic Explanatory Interactive Learning

The Color-MNIST example clearly shows that although the input-level explanations of current XAI methods are an important first step towards true explanations of a model's behavior, a large amount of ambiguity in a model's decision process remains. Using XIL on these visual explanations only, it can be difficult to properly intervene on a model. What we require is an understandable, disentangled representation level, which the user can enquire from and intervene on.

Neuro-Symbolic Architecture. For this purpose, we construct an architecture consisting of two modules, a concept embedding and a reasoning module. The concept embedding module's task is to create a decomposed representation of the input space that can be mapped to human-understandable symbols. The task of the reasoning module is to make predictions based on this symbolic representation.

Fig. 3 gives an illustrative overview of our approach, which we formulate more precisely in the following: Given an input image $x_i \in X$, whereby $X := [x_1, ..., x_N] \in \mathbb{R}^{N \times M}$, with X being divided into subsets of N_c classes $\{X_1, ..., X_{N_c}\} \in X$ and with ground-truth class labels defined as $y \in [0, 1]^{N \times N_c}$, we have two modules, the concept embedding module, $h(x_i) = \hat{z}_i$, which receives the input sample and encodes it into a symbolic representation, with $\hat{z} \in [0,\ 1]^{N \times D}$. And the reasoning module, $g(\hat{z}_i) = \hat{y}_i$, which produces the prediction output, $\hat{y}_i \in [0, 1]^{N \times N_c}$, given the symbolic representation. The exact details of the $g(\hat{z}_i)$ and $h(x_i)$ depend on the specific implementations of these modules, and will be discussed further in sections below.

Retrieving Neuro-Symbolic Explanations. Given these two modules, we can extract explanations for the separate tasks, *i.e.*the more general input representation task and the reasoning task. We write an explanation function in a general notation as $E(m(\cdot),\ o)$, which retrieves the explanation of a specific module, $m(\cdot)$, given the module's output if it is the final module or the explanation of the previous module if it is not, both summarized as o here. For our approach, we thus have $E^g(g(\cdot),\ \hat{y}_i) =: \hat{e}_i^g$ and $E^h(h(\cdot),\ \hat{e}_i^g) =: \hat{e}_i^h$. These can represent scalars, vectors, or matrices, depending on the given

module and output. $\hat{e}_i^g$ represents the explanation of the reasoning module given the final predicted output $\hat{y}_i$, *e.g.*a logic-based rule. $\hat{e}_i^h$ presents the explanation of the concept learner given the explanation of the reasoning module $\hat{e}_i^g$, *e.g.*a visual explanation of a learned concept. In this way, the explanation of the reasoning module is passed back to the concept embedding in order to receive the explanations of the concept embedding module that contribute to the explanation of the reasoning module. This explanation pass is depicted by the gray arrows of Fig. 3. The exact definition of E^g and E^h used in this work are described below.

Revising Neuro-Symbolic Concepts. As we show in our experiments below, also Neuro-Symbolic models are prone to focusing on wrong reasons, *e.g.*confounding factors. In such a case, it is desirable for a user to intervene on the model, *e.g.*via XIL. As errors can result from different modules of the concept learner, the user must create feedback tailored to the individual module that is producing the error. A user thus receives the explanation of a module, *e.g.*$\hat{e}_i^g$, and produces an adequate feedback given knowledge of the input sample, x_i, the true class label, y_i, the model's class prediction $\hat{y}_i$ and possible internal representations, *e.g.*$\hat{z}_i$. For the user to interact with the model, the user's feedback must be mapped back into a representation space of the model.

In the case of creating feedback for a visual explanation, as in [44], [51] and [45], the mapping is quite clear: the user gives visual feedback denoting which regions of the input are relevant and which are not. This "visual" feedback is transferred to the input space in the form of binary image masks, which we denote as A_i^v.

The semantic user feedback can be in the form of relational functions, φ, for instance, *"if an image belongs to class* 1 *then one object is a large cube"*:

$$\boxed{\forall img.\ \ isclass(img, 1) \quad \Rightarrow \quad \exists obj.[in(obj, img) \ \wedge \ size(obj, large) \wedge shape(obj, cube)]\ ,}$$

We define $A_i^s := \bigvee_\varphi A_i^\varphi(\hat{z}_i \models \varphi)$ which describes the disjunction over all relational feedback functions which hold for the symbolic representation, $\hat{z}_i$, of an image, x_i.

An important characteristic of the semantic user feedback is that it can describe different levels of generalizability, so that feedback based on a single sample can be transferred to a set of multiple samples. For instance φ can hold for an individual sample, all samples of a specific class, j, or all samples of the data set. Consequently, the disjunction, $\bigvee_\varphi$, can be separated as: $A^s_{i|y_i=j} = A_i^{sample} \vee A^{class}_{c=j} \vee A^{all}$.

For the sake of simplicity, we are not formally introducing relational logic and consider the semantic feedback in tabular form. To summarize, we have the binary masks for the visual feedback $A_i^v \in [0,1]^{D^v}$ and the semantic feedback $A_i^s \in [0,1]^{D^s}$.

For the final interaction we refer to XIL with differentiable models and explanation functions, generally formulated as the explanatory loss term, $L_{expl} =$

$$\lambda_v \sum\nolimits_{i=1}^{N} r(A_i^v,\ \hat{e}_i^h\ ) + \lambda_s \sum\nolimits_{i=1}^{N} r(A_i^s,\ \hat{e}_i^g\ )\ . \qquad (1)$$

Depending on the task, the regularization function, $r(\cdot,\cdot)$, can be the *RRR* term of Ross *et al.* [44] or the *HINT* term of Selvaraju *et al.* [48]. The hyperparameters λ_v and λ_s control how much the different feedback forms are taken into account. Finally, the explanatory loss is concatenated to the original task dependent loss term, *e.g.*the cross-entropy loss for a classification task.

Reasoning Module. As the output of our concept embedding module represents an unordered set, whose class membership is unaltered by the order of the objects within the set, we require our reasoning module to handle such an input structure. The Set Transformer, recently proposed by Lee *et al.* [25], is a natural choice for such a task.

To generate the explanations of the Set Transformer given the symbolic representation, $\hat{z}_i$, we make use of the gradient-based Integrated Gradients explanation method of Sundararajan *et al.* [50]. Given a function $g : \mathbb{R}^{N\times D} \to [0,1]^{N\times C}$ the Integrated Gradients method estimates the importance of the jth element from an input sample z_i, z_{ij}, for a model's prediction by integrating the gradients of $g(\cdot)$ along a straight line path from z_{ij} to a baseline input, $\tilde{z}$, as:

$$IG_j(z_i) := (z_{ij} - \tilde{z}) \times \int_{\alpha=0}^{1} \frac{\delta\, g(\tilde{z} + \alpha \times (z_i - \tilde{z}))}{\delta z_{ij}} \delta\alpha.$$

Given the input to the Set Transformer, $\hat{z} \in [0,\ 1]^{N\times D}$, and $\tilde{z} = 0$ as a baseline input, we finally apply a zero threshold to only receive positive importance and thus have:

$$\hat{e}_i^h := \sum_{j=1}^{D} min(IG_j(\hat{z}_i),\ 0)\ . \qquad (2)$$

(Slot) Attention is All You Need (for object-based explanations). Previous work of Yi *et al.* [58] and Mao *et al.* [32] has shown an interesting approach for creating a Neuro-Symbolic concept leaner based on a Mask-RCNN [11] scene parser. For our concept learner, we make use of the recent work of Locatello *et al.* [29]. Their proposed Slot Attention module allows to decompose the hidden representation of an encoding space into a set of task-dependent output vectors, called "slots". For example, the image encoding of a CNN backbone can be decomposed such that the hidden representation is separated into individual slots for each object. These decomposed slot encodings can then be used in further task-dependent steps, *e.g.*attribute prediction of each object. Thus with Slot Attention, it is possible to create a fully differentiable object-centric representation of an entire image without the need to process each object of the scene individually in contrast to the system of [58, 32].

An additional important feature of the Slot Attention module for our setting is the ability to map each slot to the original input space via the attention maps. These attention maps are thus natural, intrinsic visual explanations of the detected objects. In contrast, with the scene parser of [58, 32] it is not as straightforward to generate visual explanations based on the explanations of the reasoning module. Consequently, using the Slot Attention module, we can formulate the dot-product

Data Set	Size	Classes	Input-dimensions	Multi-object	Visual confounder	Non-visual confounder	Number of ruletypes
ToyColor [44]	40k	2	$5 \times 5 \times 3$	✗	✓	✗	1
ColorMNIST [17]	70k	10	$28 \times 28 \times 3$	✗	✗	✓	1
Decoy-MNIST [44]	70k	10	$28 \times 28 \times 3$	✗	✓	✗	1
Plant Data Set [45]	2.4k	2	$213 \times 213 \times 64$	✗	✓	✗	1
ISIC Skin Cancer Data Set [4, 52]	21k	2	$650 \times 450 \times 3$	✗	✓	✗	1
Our CLEVR-Hans3	13.5k	3	$320 \times 480 \times 3$	✓	✓	✓	2
Our CLEVR-Hans7	31.5k	7	$320 \times 480 \times 3$	✓	✓	✓	4

Table 1: **The complexity of CLEVR-Hans.** The CLEVR-Hans data sets represent confounded data sets in which the confounding factors are not separable in the original input space. Additionally, more than one global class rules must be applied in order to revise the model.

attention for a sample x_i, as

$$B_i := \sigma\left(\frac{1}{\sqrt{D'}}\, k(F_i) \cdot q(S_i)^T\right) \in \mathbb{R}^{P \times K}\,,$$

where σ is the softmax function over the slots dimension, $k(F_i) \in \mathbb{R}^{P \times D'}$ a linear projection of the feature maps F_i of an image encoder for x_i, $q(S_i) \in \mathbb{R}^{K \times D'}$ a linear projection of the slot encodings S_i and $\sqrt{D'}$ a fixed softmax temperature. P represents the feature map dimensions, K the number of slots and D' the dimension which the key and query functions map to.

Finally, we can formulate $E^h(h(\cdot),\ \hat{e}_i^g)$ of Eq. 4 based on the attention maps B_i, and the symbolic explanation $\hat{e}_i^h$. Specifically, we only want an explanation for objects which were identified by the reasoning module as being relevant for the final prediction:

$$\hat{e}_i^h := \sum_{k=1}^{K} \begin{cases} B_{ik}, & \text{if } max(\hat{e}_{ik}^g) \geq t \\ \mathbf{0} \in \mathbb{R}^P, & \text{otherwise} \end{cases}\,, \tag{3}$$

where t is a pre-defined importance threshold. Alternatively the user can manually select explanations for each object.

Interchangeability of the Modules. Though both Slot-Attention and Set Transformer have strong advantages as stated above, alternatives exist. Deep Set Prediction Networks [60], Transformer Set Prediction Networks [21] or Mask-RCNN based models [11] are viable alternatives to the Slot Attention module as concept embedding module. The generation of visual explanations within these models, *e.g.*via gradient-based explanation methods, however, is not as straightforward. Truly rule-based classifiers [39, 30], logic circuits [27], or probabilistic approaches [5, 38, 20, 31], are principally viable alternatives for the Set Transformer, though it remains preferable for this module to handle unordered sets.

5 The CLEVR-Hans Data Set

Several confounded computer vision data sets with varying properties, *e.g.*number of classes, already exist. Tab. 1 provides a brief overview of such data sets. We distinguish here between the number of samples, number of classes, image dimensions, and whether an image contains multiple objects. More important are whether a confounding factor is spatially separable from the relevant features, *e.g.*the colored corner spots in Decoy-MNIST, whether the confounding factor is not visually separable, *e.g.*the color in ColorMNIST that superimposes the actual digits, and finally, once the confounding factor has been identified, how many different global class rules must be applied in order to revise the model, *i.e.*the corner rule for the digits in Decoy-MNIST is the same, regardless of which specific class is being considered.

To the best of our knowledge, the confounded data sets listed in Tab.1, apart from ColorMNIST, possess spatially separable confounders. One can, therefore, revise a model by updating its spatial focus. However, this is not possible if the confounding and true factors are not so easily separable in the input dimensions.

The CLEVR data set of [14] is a particularly interesting data set, as it was originally designed to diagnose reasoning modules and presents complex scenes consisting of multiple objects and different relationships between these objects. Using the available framework of [14], we have thus created a new confounded data set, which we refer to as the CLEVR-Hans data set. This data set consists of CLEVR images divided into several classes. The membership of a class is based on combinations of objects' attributes and relations. Additionally, certain classes within the data set are confounded. Thus, within the data set, consisting of train, validation, and test splits, all train, and validation images of confounded classes will be confounded with a specific attribute or combination.

We have created two variants of this data set, which we refer to as CLEVR-Hans3 and CLEVR-Hans7. CLEVR-Hans3 contains three classes, of which two are confounded. Fig. 4 shows a schematic representation of this data set. Images of the first class contain a large cube and large cylinder. The large cube has the color gray in every image of the train and validation set. Within the test set, the color of the large cube is shuffled randomly. Images of the second class contain a small sphere and small metal cube. The small sphere is made of metal in all training and validation set images, however, can be made of either rubber or metal in the test set. Images of the third class contain a large blue sphere and a small yellow sphere in all images of the data set. This class is not confounded. CLEVR-Hans7 contains seven classes, of which four are confounded. This data set, next to containing more class rules, also contains more complex class rules than CLEVR-Hans3, *e.g.*class rules are also based on object positions. For more details, we refer to the Supplementary Materials.

Finally, the images were created such that the exact combinations of the class rules did not occur in images of other classes. It is possible that a subset of objects from one class

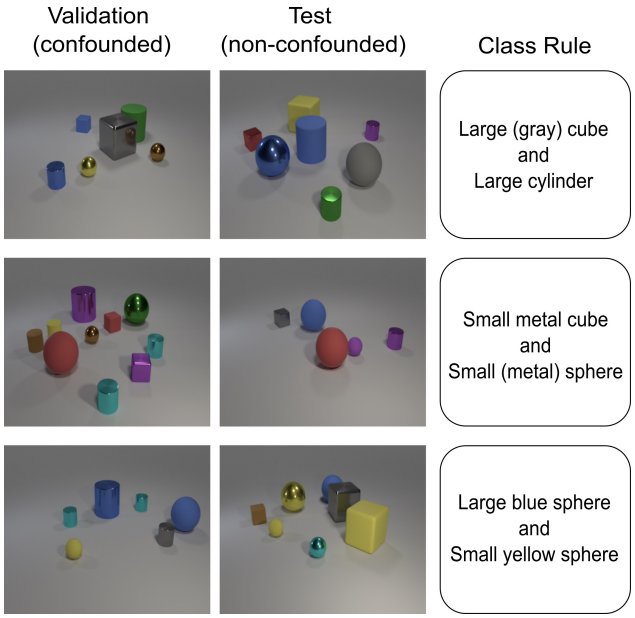

Figure 4: **Schematic of the CLEVR-Hans3 data set.** Confounding attributes are in brackets.

rule occur in an image of another class. However, it is not possible that more than one complete class rule is contained in an image. In summary, these data sets present an opportunity to investigate confounders and model decisions for complex classification rules within a benchmark data set that is more complex than previously established confounded data sets (see Tab. 1).

6 Experimental Evidence

Our intention here is to investigate the benefits of Neuro-Symbolic Explanatory Interactive Learning. To this end, we make use of our CLEVR-Hans data sets to investigate (1) the downsides of deep learning (DL) models in combination with current (visual) XAI methods and, in comparison, (2) the advantages of our NeSy XIL approach. In particular, we intend to investigate the benefits of neuro-symbolic explanations to not just provide more detailed insights of the learned concept, but allow for better interaction between human users and the model's explanations. We present qualitative as well as quantitative results for each experiment. Additional details on the experiments and our implementation, and additional qualitative results, can be found in the supplement.

Architectures. We compared our Neuro-Symbolic architecture to a ResNet-based CNN model [12], which we denote as CNN. For creating explanations of the CNN, we used the Grad-CAM method of Selvaraju *et al.* [47], a back-propagation based explanation method that visualizes the gradients of the last hidden layer of the network's encoder, and represents a trade-off between high visual representation and spatial information.

Due to the modular structure of our Neuro-Symbolic concept learner, Clever-Hans behavior can be due to errors within its sub-modules. As previous work [44, 48, 51, 45] has already shown how to revise visual explanations, we did not focus on revising the visual explanations of the concept learner for our experiments. Instead, we assumed the concept embedding module to produce near-perfect predictions and visual explanations and focused on revising the higher-level explanations of the reasoning module. Therefore, we employed a Slot-Attention module pre-trained on the original CLEVR data set [29].

Preprocessing. We used the same pre-processing steps as the authors of the Slot-Attention module [29].

Training Settings. We trained the two models using two settings: A standard classification setting using the cross-entropy loss (Default) and the XIL setting where the explanatory loss term (Eq. 1) was appended to the cross-entropy term. The exact loss terms used will be discussed in the corresponding subsections.

User Feedback. As in [51, 48, 45], we simulated the user feedback. The exact details for each experiment can be found in the corresponding subsections.

Evaluation. Apart from qualitatively investigating the explanations of the models, we used the classification accuracy on the validation and test set as an indication of a model's ability to make predictions based on correct reasons. If the accuracy is high on the confounded validation set but low on the non-confounded test set, it is fair to assume that the model focuses on the confounding factors of the data set to achieve a high validation accuracy. We ran all experiments with five random parameter initializations and report the mean classification accuracy with standard deviation over these runs.

6.1 Visual XIL fails on CLEVR-Hans

We first demonstrate the results of training a standard CNN for classification.

CNN produces Clever-Hans moment. As Tab. 2 indicates, the default CNN is prone to focusing on the confounding factors of the data sets. It reaches near perfect classification accuracies in the confounded validation sets but much lower accuracy in the non-confounded test sets. Interestingly, the main difficulty of the standard CNN for CLEVR-Hans3 appears to lie in the gray color confounder of class 1, whereas the confounding material of class 2 does not appear to be a difficulty for the model (*cf.* Supplementary Materials).

Examples of visual explanations of the default CNN for CLEVR-Hans3 images are presented in Fig. 5. Note these explanations appear rather unspecific and ambiguous, and it is not clear whether the model has learned the two object class rules of CLEVR-Hans3.

Revising Visual Explanations via XIL. We next apply XIL to the CNN model to improve its explanations. As in [48, 45] we set $r(A^v, \hat{e}^v)$ to the mean squared error between user annotation and model explanation. We simulate a user by providing ground-truth segmentation masks for each class relevant object

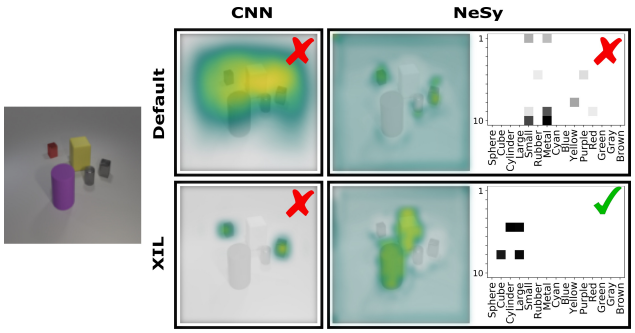

Figure 5: **Example explanations (from test set) of different model and training settings on CLEVR-Hans3.** Red crosses denote false, green checks correct predictions.

in the train set. In this way, we could improve the model's explanations to focus more on the relevant objects of the scene.

An example of the revised visual explanations of the CNN with XIL can be found in Fig. 5 again visualized via Grad-CAMs. Compared to the not revised model, one can now clearly detect which objects are relevant for the model's prediction. However, the model's learned concept seems to not agree with the correct class rule, *cf.* Fig. 4, and thus, in this case, it is not able to predict the correct class. Further, it is still ambiguous what concepts about those objects are relevant for the model. The accuracies in Tab. 2 lastly indicate that correcting the visual explanations improved the overall test accuracy by a margin, however comparing to the near-perfect validation accuracy, it is clear the model still focuses on confounding factors.

6.2 Neuro-Symbolic XIL to the Rescue

Now, we are ready to investigate how Neuro-Symbolic XIL improves upon visual XIL.

Receiving Explanations of Neuro-Symbolic model. Training the Neuro-Symbolic model in the default cross-entropy setting, we make two observations. Firstly, we can observe an increased test accuracy compared to the previous standard CNN settings. This is likely due to the class rules' relevant features now being more evident for the model to use than the standard CNN could possibly catch on to, *e.g.*the object's material. Secondly, even with a higher test accuracy than the previous model could achieve, this accuracy is still considerably below the again near perfect validation accuracy. This indicates that also the Neuro-Symbolic model is not resilient against confounding factors.

Example explanations of the Neuro-Symbolic model can be found in Fig. 5, with the symbolic explanation on the right side and the corresponding attention-based visual explanation left of this. The objects highlighted by the visual explanations depict those objects that are considered as most relevant according to the symbolic explanation (see Eq. 3 for details). These visualizations support the observation that the model also focuses on confounding factors.

Revising Neuro-Symbolic Models via Interacting with Their Explanations. We observe that the Clever-Hans moment of the Neuro-Symbolic model in the previous experiment was mainly due to errors of the reasoning module as the visual explanation correctly depicts the objects that were considered as relevant by the reasoning module. To revise the model we therefore applied XIL to the symbolic explanations via the previously used, mean-squared error regularization term. We provided the true class rules as semantic user feedback.

Model	Validation (confounded)	Test (non-confounded)
CLEVR-Hans3		
CNN (Default)	99.55 ± 0.10	70.34 ± 0.30
CNN (XIL)	99.69 ± 0.08	70.77 ± 0.37
NeSy (Default)	98.55 ± 0.27	○ $\mathbf{81.71 \pm 3.09}$
NeSy XIL	100.00 ± 0.00	• $\mathbf{91.31 \pm 3.13}$
CLEVR-Hans7		
CNN (Default)	96.09 ± 0.19	84.50 ± 1.04
CNN (XIL)	96.08 ± 0.25	89.26 ± 0.29
NeSy (Default)	96.88 ± 0.16	○ $\mathbf{90.97 \pm 0.91}$
NeSy XIL	98.76 ± 0.17	• $\mathbf{94.96 \pm 0.49}$

CLEVR-Hans3 – Global Correction Rule (¬Gray)

Model	Test (class 1)	Test (all classes)
NeSy (Default)	52.98 ± 9.60	81.71 ± 3.09
NeSy XIL	$\mathbf{83.59 \pm 8.44}$	$\mathbf{83.26 \pm 6.46}$

Table 2: **Accuracies on Clevr-Hans3 and Clevr-Hans7**. The best ("•") and runner-up ("○") results are bold. We compare the test accuracy in comparison to the validation accuracy as an indication of Clever-Hans moments.

The resulting accuracies of the revised Neuro-Symbolic model can be found in Tab. 2 and example explanations in Fig. 5. We observe that false behaviors based on confounding factors could largely be corrected. The XIL revised Neuro-Symbolic model produces test accuracies much higher than was previously possible in all other settings, including the XIL revised CNN.

To test the influence of possible Slot-Attention prediction errors we also tested revising the reasoning module when given the ground-truth symbolic representations. Indeed this way, the model could reach a near-perfect test accuracy (*cf.* Supplementary Materials).

Revision via General Feedback Rules. Using XIL for revising a model's explanations requires that a human user interacts with the model on a sample-based level, *i.e.*the user receives a model's explanation for an individual sample and decides whether the explanation for this is acceptable or a correction on the model's explanation is necessary. This can be very tedious if a correction is not generalizable to multiple samples and must thus be created for each sample individually.

Consider class 1 of CLEVR-Hans3, where the confounding factor is the color gray of the large cube. Once gray has been identified as an irrelevant factor for this, but also all other classes, using NeSy XIL, a user can create a global correction

rule as in Fig. 3. In other words, irrespective of the class label of a sample, the color gray should never play a role for prediction.

Tab. 2(bottom) shows the test accuracies of our Neuro-Symbolic architecture for class 1 and, separately, over all classes. We here compare the default training mode vs. XIL with the single global correction rule. For this experiment, our explanatory loss was the RRR term [44] which has the advantage of handling negative user feedback.

We observe that applying the correction rule has substantial advantages for class 1 test accuracies and minor advantages for the full test accuracy. These results highlight the advantage of NeSy XIL for correcting possible Clever-Hans moments via global correction rules, a previously non-trivial feature.

7 Conclusion

Neuro-Symbolic concept learners are capable of learning visual concepts by jointly understanding vision and symbolic language. However, although they combine system 1 and system 2 [15] characteristics, their complexity still makes them difficult to trust in critical applications, especially, as we have shown, if the training conditions for their system 1 component may differ from those in the test condition. However, their system 2 component allows one to identify when models are right for the wrong conceptual reasons. This allowed us to introduce the first Neuro-Symbolic Explanatory Interactive Learning approach, regularizing a model by examining and selectively penalizing its Neuro-Symbolic explanations. The results of our empirical evaluation on a newly compiled confounded benchmark data set, called CLEVR-Hans, demonstrated that semantic explanations, i.e., compositional explanations at a per-object, symbolic level, can identify confounders that are not identifiable using "visual" explanations only. More importantly, feedback on this semantic level makes it possible to revise the model from focusing on these confounding factors.

Our results show that Neuro-Symbolic explanations and interactions merit further investigation. Using a semantic loss [57] would allow one to stay at the conceptual level directly. Furthermore, one should integrate a neural semantic parsing system that helps to interactively learn a joint symbolic language between the machine and the human user through decomposition [16]. Lastly, language-guided XIL [35] is an interesting approach for more natural supervision. These approaches would help to move from XIL to conversational XIL. Applying Neuro-Symbolic prior knowledge to a model may provide additional benefits to a XIL setting. Finally, it is very interesting to explore more expressive reasoning components and investigate how they help combat even more complex Clever-Hans moments. Concerning our data set, an interesting next step would be to create a confounded causal data set in the approach of [10].

Ackowledgements

We acknowledge the support by BMEL/BLE funds under the innovation support program, project "AuDiSens" (FKZ 28151NA187). We additionally wish to thank Thomas Kipf for support with Slot Attention.

References

[1] Somak Aditya, Yezhou Yang, and Chitta Baral. Explicit reasoning over end-to-end neural architectures for visual question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI), the 30th innovative Applications of Artificial Intelligence (IAAI), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI), 2018*, pages 629–637. AAAI Press, 2018.

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR, 2018*. OpenReview.net, 2018.

[3] Gabriele Ciravegna, Francesco Giannini, Marco Gori, Marco Maggini, and Stefano Melacci. Human-driven FOL explanations of deep learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 2234–2240. ijcai.org, 2020.

[4] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[5] Adnan Darwiche. A differential approach to inference in bayesian networks. *J. ACM*, 50(3):280–305, 2003.

[6] Artur S. d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6(4):611–632, 2019.

[7] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[8] Artur S d'Avila Garcez, Krysia B Broda, and Dov M Gabbay. *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media, 2012.

[9] Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. Towards automatic concept-based explanations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 9273–9282, 2019.

[10] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for compositional actions & temporal reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV*, pages 2980–2988. IEEE Computer Society, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778. IEEE Computer Society, 2016.

[13] Drew A. Hudson and Christopher D. Manning. Learning by abstraction: The neural state machine. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-*

14 December 2019, Vancouver, BC, Canada, pages 5901–5914, 2019.

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2901–2910, 2017.

[15] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[16] Siddharth Karamcheti, Dorsa Sadigh, and Percy Liang. Learning adaptive language interfaces through decomposition. *arXiv*, abs/2010.05190, 2020.

[17] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9012–9020, 2019.

[18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR, 2018.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[20] Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

[21] Adam R Kosiorek, Hyunjik Kim, and Danilo J Rezende. Conditional set generation with transformers. *arXiv preprint arXiv:2006.16841*, 2020.

[22] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, pages 951–958. IEEE Computer Society, 2009.

[23] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

[24] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[25] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning, ICML*, pages 3744–3753. PMLR, 2019.

[26] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI), the 30th innovative Applications of Artificial Intelligence (IAAI), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI), 2018*, pages 3530–3537. AAAI Press, 2018.

[27] Yitao Liang and Guy Van den Broeck. Learning logistic circuits. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4277–4286. AAAI Press, 2019.

[28] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L. Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4185–4194. Computer Vision Foundation / IEEE, 2019.

[29] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020.

[30] Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

[31] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 3753–3763, 2018.

[32] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019.

[33] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4942–4950. IEEE Computer Society, 2018.

[34] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[35] Jesse Mu, Percy Liang, and Noah Goodman. Shaping visual representations with language for few-shot classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4823–4830, Online, July 2020. Association for Computational Linguistics.

[36] W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*, 2018.

[37] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8779–8788. IEEE Computer Society, 2018.

[38] Hoifung Poon and Pedro M. Domingos. Sum-product networks: A new deep architecture. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pages 689–690. IEEE Computer Society, 2011.

[39] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[40] Johannes Rabold, Hannah Deininger, Michael Siebers, and Ute Schmid. Enriching visual with verbal explanations for relational concepts–combining lime with aleph. In *Joint European*

Conference on Machine Learning and Knowledge Discovery in Databases PKDD/ECML , pages 180–192. Springer, 2019.

[41] Nazneen Fatema Rajani, Rui Zhang, Yi Chern Tan, Stephan Zheng, Jeremy Weiss, Aadit Vyas, Abhijit Gupta, Caiming Xiong, Richard Socher, and Dragomir R. Radev. ESPRIT: explaining solutions to physical reasoning tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, 2020*, pages 7906–7917. Association for Computational Linguistics, 2020.

[42] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[43] Laura Rieger, Chandan Singh, W James Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *Proceedings of International Conference on Machine Learning, ICML*. PMLR, 2020.

[44] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of International Joint Conference on Artificial Intelligence IJCAI*, pages 2662–2670, 2017.

[45] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[46] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2020.

[47] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV*, pages 618–626. IEEE Computer Society, 2017.

[48] Ramprasaath Ramasamy Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry P. Heck, Dhruv Batra, and Devi Parikh. Taking a HINT: leveraging explanations to make vision and language models more grounded. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2591–2600. IEEE, 2019.

[49] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[50] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70, pages 3319–3328. PMLR, 2017.

[51] Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society AIES*, pages 239–245, 2019.

[52] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5:180161, 2018.

[53] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and D. Parikh. Probabilistic neural-symbolic models for interpretable visual question answering. In *Proceedings of International Conference on Machine Learning, ICML*, 2019.

[54] Daniel S. Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Commun. ACM*, 62(6):70–79, 2019.

[55] Jialin Wu and Raymond J. Mooney. Self-critical reasoning for robust visual question answering. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 8601–8611, 2019.

[56] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020.

[57] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR, 2018.

[58] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 1039–1050, 2018.

[59] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[60] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Deep set prediction networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 3207–3217, 2019.

[61] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

Supplementary Material

CLEVR-Hans data set

For CLEVR-Hans classes for which class rules contain more than three objects, the number of objects to be placed per scene was randomly chosen between the minimal required number of objects for that class and ten, rather than between three and ten, as in the original CLEVR data set.

Each class is represented by 3000 training images, 750 validation images, and 750 test images. The training, validation, and test set splits contain 9000, 2250, and 2250 samples, respectively, for CLEVR-Hans3 and 21000, 5250, and 5250 samples for CLEVR-Hans7. The class distribution is balanced for all data splits.

CLEVR-Hans7 The first, second, and seventh class rules of CLEVR-Hans7 correspond to classes one, two, and three of CLEVR-Hans3. Images of the third class of CLEVR-Hans7 contain a small cyan object in front of two red objects. The cyan object is a small cube in all images of the training and validation set, yet it can be any shape and size within the test set. Images of the fourth class contain at least five small objects. One of these must be green, one brown, and one purple. There are no constraints on the remaining small objects. This class is not confounded. Images of class five consist of two rules. There are three spheres present in the left half of the image (class rule 5a), or there are three spheres present in the left half of the image and three metal cylinders in the right half of the image (class rule 5b). Within all data splits, including the test split, class rule 5a occurs 90% of the time and class rule 5b 10% of the time. The class rule of the sixth class is contained in class rule 5b, namely three metal cylinders in the right half of the image. This is the same for all splits.

Preprossing details We downscaled the CLEVR-Hans images to visual dimensions 128 x 128 and normalized the images to lie between -1 and 1. For training the Slot-Attention module, an object is represented as a vector of binary values for the shape, size, color, and material attributes and continuous values between 0 and 1 for the x, y, and z positions. We refer to [29] for more details.

ColorMNIST Experiment

The model used for the ColorMNIST data set is described in Tab 3.

This model was trained with an initial learning rate of 1.0 for 14 epochs with a batch size of 64 using a step learning rate scheduler with step size 1 and $\gamma = 0.7$ and Adadelta [59] as optimizer.

Model Details and Hyperparameters

Cross-validation For cross-validating over different random parameter initializations, we used the seeds: 0, 1, 2, 3, 4.

Reasoning Module For our reasoning module, we used the recently proposed Set Transformer, an attention-based neural network designed to handle unordered sets. Our implementation consists of two stacked Set Attention Blocks (SAB) as encoder and a Pooling by Multihead Attention (PMA) decoder. Architecture details can be found in Tab 4

Concept Embedding Module For our concept embedding module, we used the set prediction architecture of Locatello *et al.* [29] that the authors had used for the experiments on the original CLEVR data set. We refer to their paper for architecture parameters and details rather than duplicating these here.

We pre-trained this set prediction architecture on the original CLEVR data set with a cosine annealing learning rate scheduler for 2000 epochs, minimum learning rate $1e-5$, initial learning rate $4e-4$, batch size 512, 10 slots, 3 internal slot-attention iterations and the Adam optimizer [19] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\epsilon = 1e-08$ and zero weight decay.

Type	Size/Channels	Activation	Comment
Conv 3 x 3	32	ReLu	stride 1
Conv 3 x 3	64	ReLu	stride 1
AdaptiveAvgPool (2D)	14×14	-	-
Dropout	-	-	p = 0.25
Flatten	-	-	dim = 1
Linear	128	-	-
Dropout	-	-	p = 0.5
Linear	10	-	-

Table 3: CNN used for ColorMNIST experiments.

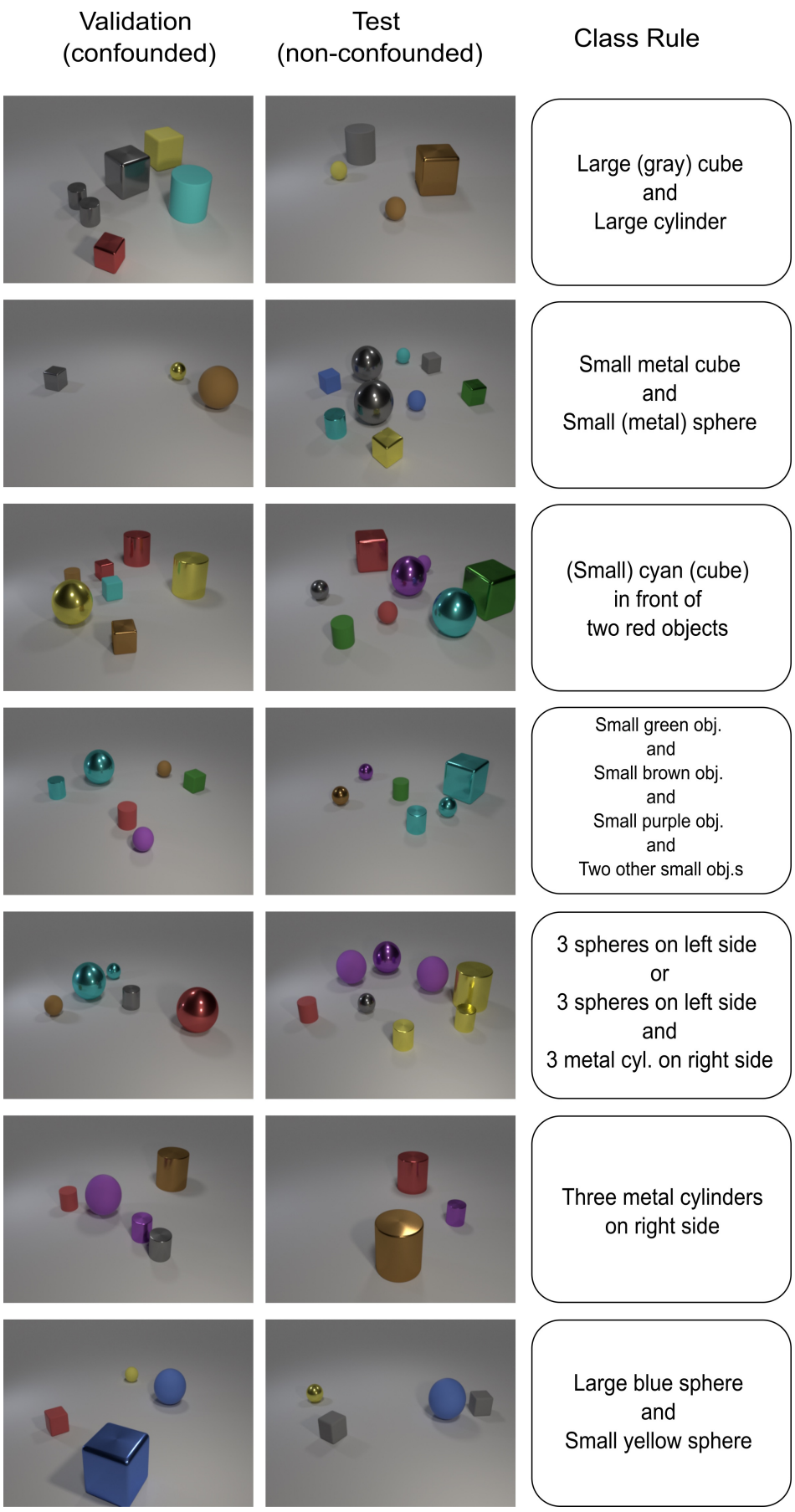

Figure 6: **CLEVR-Hans7 data set overview.** Please refer to the main text for a more detailed description of the data set.

Type	Dim Out	Numb. Heads	Comment
SAB	128	4	-
SAB	128	4	-
Dropout	-	-	p = 0.5
PMA	128	4	-
Dropout	-	-	p = 0.5
Linear	3/7	-	-

Table 4: Set Transformer architecture used for reasoning module. Depending on whether CLEVR-Hans3 or CLEVR-Hans7 was used the final output varied between 3 and 7.

Neuro-Symbolic Concept Learner To summarize, we thus have the two modules, as stated above. For our experiments, we passed an image through the pre-trained concept embedding module. For simplicity, we binarized the output of the concept embedding module for the attributes shape, size, and color, before passing it to the reasoning module. This way, each object is represented by a one-hot encoding of each of these attributes.

The architecture parameters of the concept embedding and reasoning module were as stated above, and the same for both training settings, i.e., default and XIL.

In the default training setting, using the cross-entropy classification loss, we used the Adam optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\epsilon = 1e-08$ and zero weight decay) in combination with a cosine annealing learning rate scheduler with initial learning rate $1e-4$, minimal learning rate $1e-6$, 50 epochs and batch size of 128.

For training our concept learner using the HINT [48] loss term on the symbolic explanations in addition to cross entropy term we used the Adam optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\epsilon = 1e-08$ and zero weight decay) in combination with a cosine annealing learning rate scheduler with initial learning rate $1e-3$, minimal learning rate $1e-6$, 50 epochs and batch size of 128. We used $\lambda_s = 1000$ for the XIL experiments on CLEVR-Hans3 and $\lambda_s = 10$ for the XIL experiments on CLEVR-Hans7. For the global rule experiments, using the RRR term of Ross et al. [44], we set $\lambda_s = 20$ with all other hyperparameters the same as previously.

CNN Model Details Our CNN model is based on the popular ResNet34 model of [12]. The visual explanations generated by Grad-CAM are in the visual dimensions of the hidden feature maps. As these dimensions of the ResNet34 model were very coarse given our data pre-processing, we decreased the number of layers of the ResNet34 model by removing the last six convolutional layers (i.e., fourth of the four ResNet blocks) and adjusting the final linear layer accordingly.

For training the CNN in default cross-entropy mode, we used a constant learning rate of $1e-4$ for 100 epochs and a batch size of 64. We used the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\epsilon = 1e-08$ and zero weight decay. For training the CNN with an additional HINT explanation regularization, we used the same training parameters, as in the default case, and a $\lambda_v = 10$. These parameters were the same for CLEVR-Hans3 and CLEVR-Hans7.

Explanation Loss Terms

For our experiments, we used two different types of explanation loss terms (Eq. 4). For all experiments, apart from those with a single global rule, we simulated the user feedback as positive feedback. In other words, the user feedback indicated what features the model should be focusing on. For simplicity in our experiments, we simulated the user to have full knowledge of the task and give the fully correct rules or visual regions as feedback. For this positive feedback, we applied a simple mean-squared error between the model explanations and user feedback as an explanation loss term:

$$L(\theta,\ X,\ y,\ A) = \lambda_1 \frac{1}{N} \sum_{i=1}^{N} \sum_{d=1}^{D} \left(A_{id} - \hat{e}^g_{id}\right)^2 \tag{4}$$

This was applied to the XIL experiments with the standard CNN model, for which the explanations were in the form of Grad-CAMs, and for revising the Neuro-Symbolic model. In the case of revising the CNNs, the user annotation masks were downscaled to match the Grad-CAM size resulting from the last hidden layer of the CNN.

For handling the negative feedback of the experiments with the single global rule, in which the user indicated which features are not relevant, rather than which are, we reverted to the RRR term of Ross et al. [44]:

$$L(\theta,\ X,\ y,\ A) = \lambda_1 \sum_{i=1}^{N} \sum_{d=1}^{D} \left(A_{id} \frac{\delta}{\delta \hat{z}_{id}} \sum_{k=1}^{N_c} \log(\hat{y}_{ik}) \right)^2 \tag{5}$$

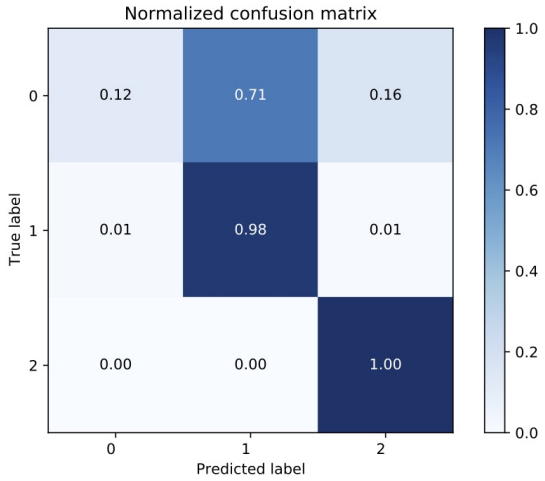

Figure 7: **Confusion matrix of CNN with cross-entropy training for CLEVR-Hans3.**

Additional Explanation Visualizations

Fig. 7 presents the confusion matrix for the default CNN on the test set, indicating the model's difficulty especially with the color confounder of class on rather than the material confounder of class two.

Fig. 8 shows additional qualitative results of NeSy XIL in addition to those of the main text. The top left example (a) presents another example where only via interacting with Neuro-Symbolic explanations can get the correct prediction for the correct reason. Top right (b) shows an example where all model configurations make the correct prediction. However, it does not become clear whether the CNN is indeed focusing on both relevant objects. With the NeSy model, this becomes clearer, though only using NeSy XIL are the correct objects and attributes identified as relevant for prediction. A similar case can be found in the middle left (c), where NeSy XIL aids in focusing on both relevant objects. The middle right shows a case where already NeSy shows advantages for creating correct predictions, yet not entirely for the correct concept. The bottom example (e) exemplifies that solely from a visual explanation, it does not become clear that the model is focusing on the color confounder, gray.

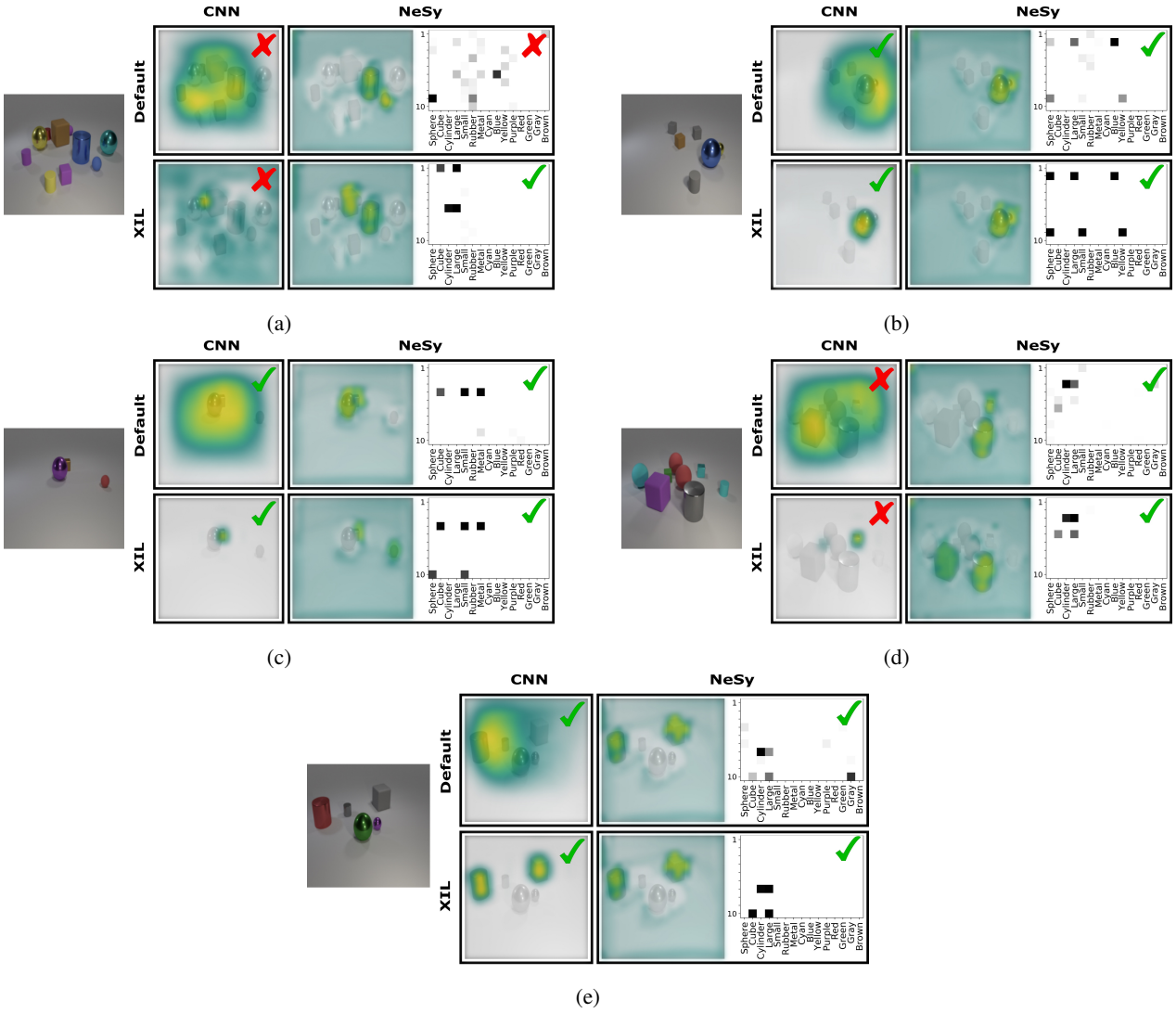

Figure 8: Additional explanations of the various model types for test samples. Green checks represent correct class predictions, red crosses incorrect predictions.