

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

## Technical Appendix

### 1 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.

**Preprocessing details** We downsampled 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 [3] for more details.

### 2 ColorMNIST Experiment

The model used for the ColorMNIST data set is described in Tab 1.

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 [6] as optimizer.

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

Table 1: CNN used for ColorMNIST experiments.

Validation (confounded)	Test (non-confounded)	Class Rule
		Large (gray) cube and Large cylinder
		Small metal cube and Small (metal) sphere
		(Small) cyan (cube) in front of two red objects
		Small green obj. and Small brown obj. and Small purple obj. and Two other small obj.s
		3 spheres on left side or 3 spheres on left side and 3 metal cyl. on right side
		Three metal cylinders on right side
		Large blue sphere and Small yellow sphere

Figure 1: **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 2: 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.

### 3 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 2

**Concept Embedding Module** For our concept embedding module, we used the set prediction architecture of Locatello *et al.* [3] 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 [2] with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ ,  $\epsilon = 1e - 08$  and zero weight decay.

**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 [5] 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. [4], 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 [1]. 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.

### 4 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

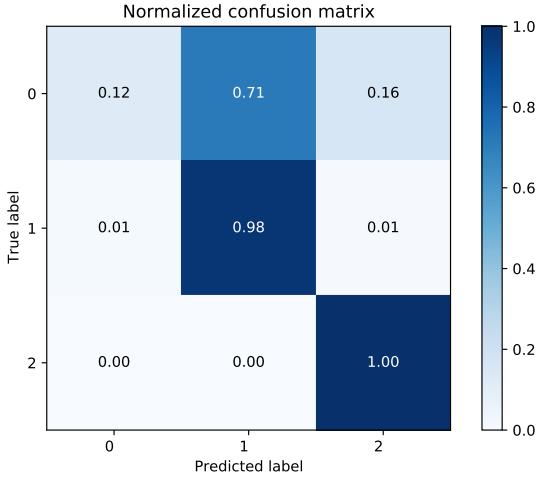


Figure 2: Confusion matrix of CNN with cross-entropy training for CLEVR-Hans3.

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:

$$L(\theta, X, y, A) = \lambda_1 \frac{1}{N} \sum_{i=1}^N \sum_{d=1}^D (A_{id} - \hat{e}_{id}^g)^2 \quad (1)$$

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 downsampled 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. [4]:

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

## 5 Additional Explanation Visualizations

Fig. 2 presents the confusion matrix for the default CNN on the test set, indicating the model's difficulty especially with the color confounder of class 0 rather than the material confounder of class 2.

Fig. 3 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.

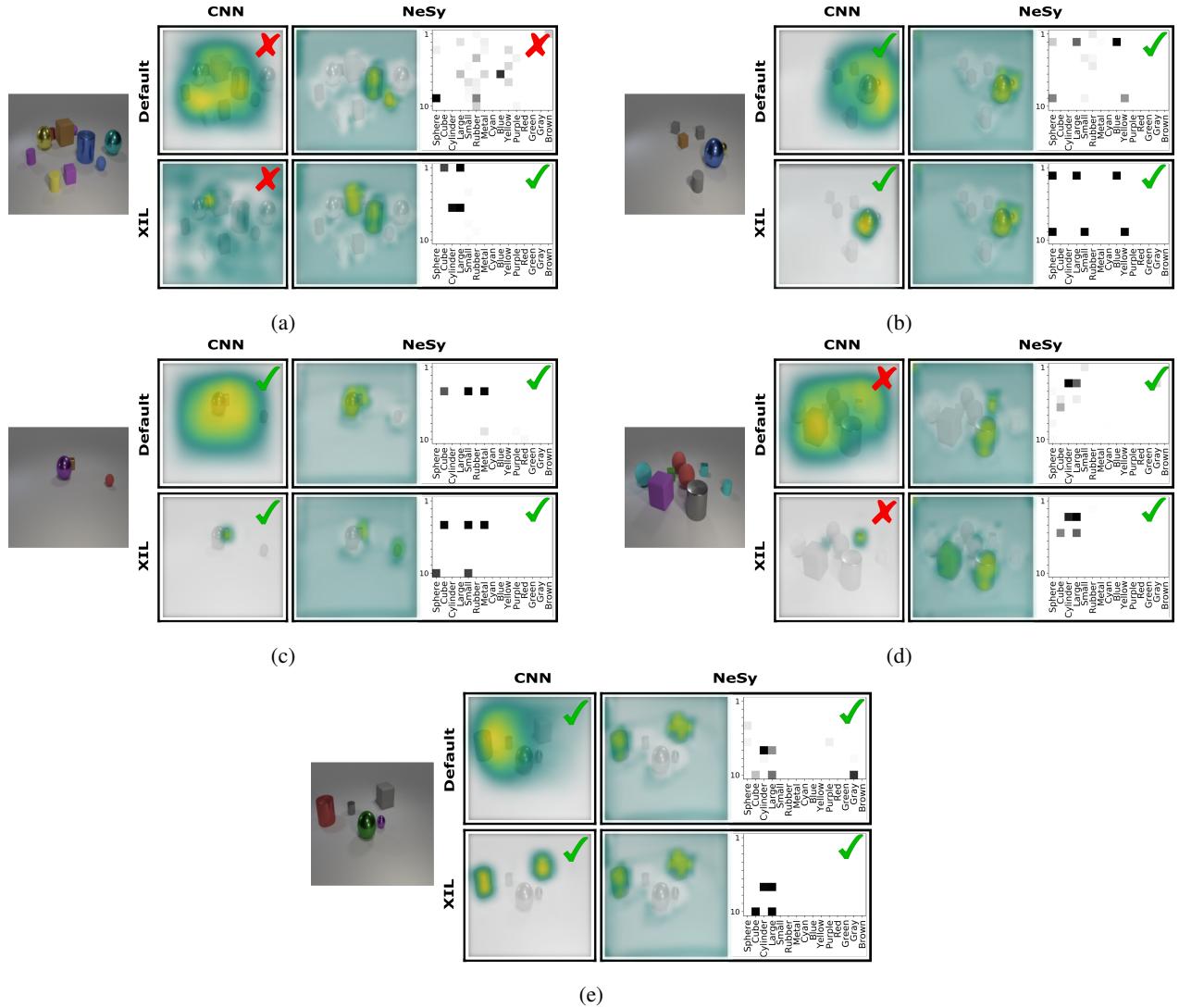


Figure 3: Additional explanations of the various model types for test samples. Green checks represent correct class predictions, red crosses incorrect predictions.

## References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.