SaSSY (Symbolic And Sub-SYmbolic) Attention: Assessing the Robustness of Neuro-Symbolic Modelling in the CLEVR-Hans3 Dataset: Final Report

Xuelong An, Paolo Cassina, Jingxuan Chen

Abstract

As a branch of Artificial Intelligence, Neuro-Symbolic (NeSy) AI aims to create models capable of robust and parsimonious learning. In our study, we expand on work by Stammer et al. (2021) to test the robustness and parsimony of traditional convolutional neural networks (CNN) and Neuro-Symbolic (NeSy) architectures comprising a Slot Attention and a Set Transformer component. We evaluate their robustness by comparing their classification accuracy after fine-tuning them to a modified version of the CLEVR-Hans3 dataset containing four different kinds of data complications. We find that models using the Slot Attention maintained good classification performance across data complications, indicating that the object-centric representations built by this perceptual component are crucial for model robustness. While a Slot Attention + ResNet18 architecture had the best overall performance, we point out that Slot Attention + Set Transformer is much more parsimonious since it achieves similar results with 20 times fewer parameters.

1. Introduction

Convolutional neural networks (CNN) have led to breakthroughs in image classification tasks. However, they have been recognised to present significant problems (Garcez & Lamb, 2020).

Neural networks store the patterns learned from training data in a distributed manner (Theodoridis, 2020). This is a defining attribute of these models' ability to learn countless features in the data, but it also means that it is challenging to chart out what concepts are being learned during training, and hence diagnose where learning "went wrong" if the model does not behave as expected at test time. For instance, deep neural networks have been documented to be vulnerable to the presence of confounders in the task of image classification. When a deep neural network is affected by a confounder during training, it shows unexpected behaviour in that it learns the wrong aspect of the data (see Figure 1 for an example). The issue of confounders draws attention to the general drawback of these models – low interpretability: the decisions taken by deep learning models are hard for humans to interpret and thus diagnose. Explanation methods for deep learning often consist of mapping importance estimates for a model's prediction to the original input space (Du et al., 2019). Nevertheless, low explainability remains a central challenge for deep learning models (Stammer et al., 2021).

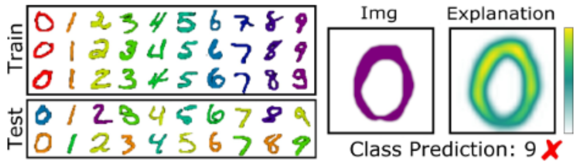

Figure 1. Illustration of the effect of confounding on a traditional CNN, for the task of digit classification. If by chance the training data distribution is heavily skewed towards a colour for each digit, colour acts as a confounder, since it induces the model to make an unwarranted generalisation from digit colour to digit class. In this case, a traditional CNN would learn to predict class '9' from the colour 'purple'. As a result, at test time the model misclassifies a purple 0 as a 9 with high confidence. In this case, the visual explanation for CNN's wrong prediction, based on a feature map, is uninformative of the model's behaviours. Figure extracted from (Stammer et al., 2021).

The lack of parsimony has also been identified as a drawback of deep neural networks (Garcez & Lamb, 2020; Marcus, 2020). These models are computationally expensive, with standard architectures counting tens of millions of parameters, and often need very large amounts of high-quality training data in order to effectively tune their parameters and perform well with out-of-distribution data points.

Given these problems, NeSy AI seeks to integrate neural and symbolic AI architectures, with the objective of accomplishing robust and parsimonious learning as well as sound higher-level reasoning (Manhaeve et al., 2018; Garcez & Lamb, 2020). One pathway to NeSy is represented by the effort to integrate functional, task-specific, prior expert knowledge and inductive biases in the model, to make learning more interpretable, computationally tractable and parsimonious. For instance, humans are very successful in object classification tasks, and they have been shown to preferentially employ shape information (Kucker et al., 2019) when solving these. Human shape biases can be thought of as an innate toolkit that humans use to their advantage to efficiently reduce the hypothesis space when solving a task. Inspired by this, constraining learning for vision models to reproduce these biases is thought to be an effective way to improve their performance (Tuli et al., 2021). Examples of prior knowledge compiled into a neural network include the convolutional operators of a convolutional neural network to handle the shift and translational invariance of an image. Another well known inductive bias is embedded in recurrent neural networks to handle input sequences of variable length.

Tangent to the growing interest in this field, novel datasets have been proposed over the years to chart out the potential of NeSy AI. One such benchmark is CLEVR, which tests a model on the task of query-driven reasoning over an image (Johnson et al., 2017). This diagnostic dataset is thoroughly annotated as each image in the training and validation set is paired with queries and answers, along with scene graph annotations giving ground-truth locations, attributes, and relationships for objects appearing in the image. It is challenging in that multiple objects of different colours, shapes, materials and sizes can appear in a scene in varying positions and rotations, under different background illumination. Given an input scene which consists of a maximum of 10 such objects, a model is asked questions such as "how many red spheres are there?" or "what is the shape of the object next to the blue cube?". CLEVR-Hans is a variant of CLEVR proposed by Stammer et al. (2021) where each input scene is assigned a class label depending on particular specifications. It elaborates on CLEVR in order to test for generalisation under the presence of visual confounders. To perform well on CLEVR-Hans, a model has to be capable of disentangling concepts like shape from colour in order to tolerate the presence of confounders. The same authors also proposed a NeSy architecture which was able to perform significantly better than a ResNet34 baseline, arguing the advantages of the former encompassing parsimony, explainability and ability to reason over a disentangled representation of a CLEVR-Hans3 input image.

The contribution we make with this study is two-fold. Firstly, we aim to expand on Stammer et al. (2021) and further investigate the hypothesis that NeSy models offer advantages over traditional, non-hybrid CNNs with respect to parsimonious and robust learning when dealing with different kinds of real-life dataset complications. Particularly, we test robustness under distribution shift: given a model f that will be fine-tuned on the training set $(x, y) \sim \mathcal{D}$ and tested on a test set $(x', y') \sim \mathcal{D}' \neq \mathcal{D}$, we construct $\mathcal{D}$ to ensure that (x, y) and (x', y') are drawn from different distributions. Robustness is then defined around the model's performance on $\mathcal{D}'$ (Wang et al., 2021): a model f is more robust than another model f' if f is more capable of maintaining good test performance despite the complications in the training data. Informally, a robust model can be compared to a learner that has the essential capability of disentangling the crucial concepts/patterns of the data, in a way that a change of a confounding concept in the data distribution affects learning only minimally.

Having established the extent to which Nesy models are more robust and parsimonious learners, we wish to understand *what* makes NeSy offer these advantages with certain kinds of data complications. Is it the neural component, the reasoning component or an interaction of the two? We attempt to answer these questions by fine-tuning different architectures consisting of exclusively CNN components or a combination of CNN and reasoning components, and comparing test accuracy on an image classification task for the CLVER-Hans3 dataset under five complication conditions: Confounder Generalisation, Noise, Class Imbalance, Mislabelled Data and Small Training Data.

2. Motivation and Related Work

Stammer et al. (2021) compiled the CLEVR-Hans3 dataset, a confounded visual scene dataset in which each scene is classified based on specific combinations of object attributes and relations between objects. They used a NeSy model architecture, composed of a Slot Attention, which during training produces a set of object-centric abstract representations called slots; and a Set Transformer that performs inference over these representations. They showed that this architecture outperformed a ResNet-based CNN despite having 40 times fewer parameters.

Our motivation is to expand on this finding as follows. Firstly, we aim to test whether Stammer et al. (2021)'s results can be generalised to the other kinds of real-life data complications outlined below. Moreover, we run more comprehensive experiments by considering models which serve as intermediate variants transitioning from a full deep learning approach, to a NeSy approach, which helps us chart out the extent of the contribution of the neural and reasoning components of the NeSy architecture.

2.1. Dataset and Task

We employ the CLEVR-Hans3 dataset to solve a ternary image classification task. CLEVR-Hans3 is split into a training, validation and testing partition consisting of 9000, 2250 and 2250 samples respectively. Class assignment is evenly balanced in each split, e.g. in the training set, each class is represented by 3000 images. A class 0 label is assigned when an image contains at least a "large cube and large cylinder", while class 1 images contain at least "a small metal cube and small sphere" and in class 2 there is at least "a large clue sphere and a small yellow sphere". These classes are mutually exclusive and images were carefully generated so as to ensure non-overlap. By default, CLEVR-Hans3 tests a model's robustness to generalise under visual confounders, i.e., during training and validation, only images with large grey cubes and large cylinders are shown as class 0, while during testing large cubes of varying colour and large cylinders are labelled class 0, as exemplified in Figure 2. If a model is confounded by the colours of the objects in the training images, it will not be able to correctly classify images containing objects with different colours that appeared in the testing dataset.

We test the robustness of four different models (described in Section 2.2) by modifying CLEVR-Hans3, separately introducing four kinds of data complications. These are

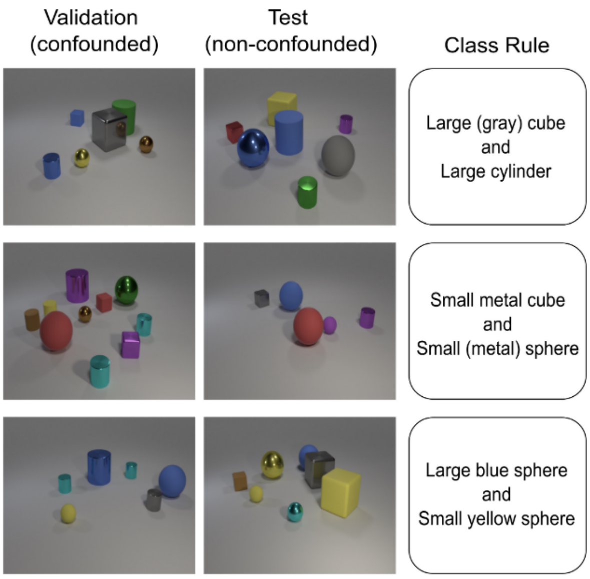
Validation (confounded)
Test (non-confounded)
Class Rule
Large (gray) cube and Large cylinder
Small metal cube and Small (metal) sphere
Large blue sphere and Small yellow sphere

Figure 2. Illustration of CLEVR-Hans3. Only Class 0 and 1 contain a visual confounder enclosed in parentheses. For instance, during the training and validation phases, only images with large (grey) cubes and small metal spheres will be shown with their respective class labels. During testing, however, the visual confounder is relaxed. For example, images of large cubes of varying colours and small metal spheres will appear labelled as class 0 in the test data. In this paper, we subject CLEVR-Hans3 to four additional complications: noise, class imbalance, small training data and mislabels. Figure extracted from (Stammer et al., 2021)

listed here together with a short rationale for each:

- **Noisy Data:** We introduce noise to the training data of CLEVR-Hans3 to simulate the complication of poor-quality data used to train a deep learning model. Imperfect and low-quality data being an omnipresent issue in the machine learning community (Redman, 2018), which can be attributed to a plethora of sources ranging from substandard measurement-collecting devices (Beede et al., 2020), difficulties with data-collection methods, or data "in the wild" being inherently noisy with a particular mention to healthcare data (Zhou et al., 2020; Miotto et al., 2018).

- **Class Imbalance:** In simulating this complication in our dataset, we are specifically inspired by the xBD Dataset (Xia et al., 2019), which is used for classifying damage severity in buildings pre- and post-natural disaster from satellite imagery. This task is made even more challenging by the data distribution being heavily skewed: intact buildings are much more common than damaged ones, due to catastrophic events being statistically infrequent. It is therefore worth exploring how our NeSy model performs amidst a heavily skewed dataset.

- **Small Training Data:** We simulate this complication because deep learning models are well-known to be data-hungry, and sometimes require exponential increases in data size (as well as the model size and computational power) to observe increases in prediction performance (Bahri et al., 2021; Kaplan et al., 2020). However, readily available big data related to a task at hand is often an unrealistic assumption. This is owed to multiple factors including the nature of the task itself such as the diagnosis of an extremely rare neurological disorder like tuberous sclerosis complex (Sánchez Fernández et al., 2020). The data annotation process can itself be demanding in terms of requiring domain-specific experts, and thus be very costly and not manageable at large scales (Brigato & Iocchi, 2021). Further motivation stems from Mao et al. (2019), who reported that by holding out 90% of the training data of the CLEVR dataset to train their NeSy model for query-answering, they were still able to defeat their CNN counterparts.

- **Mislabels:** The final condition we want to explore is the presence of mislabels during training. We motivate this condition by highlighting that supervised learning models are also over-reliant on the human annotators supplying the dataset (Biderman & Scheirer, 2020). However, annotation tasks can be tedious, leading to humans making mistakes during labelling, as well as subjective, which renders inter-annotator agreement challenging. Empirical studies consistently show that mislabels are particularly harmful to deep neural networks (Karimi et al., 2020).

2.2. Models

We proceed to explain how Stammer et al. (2021)'s NeSy model is assembled and subsequently which models we considered in our experiments. In their original paper, the NeSy model consists of a Slot Attention (Locatello et al., 2020) acting as a perception module, and a Set Transformer acting as a reasoning module (Figure 4). The Slot Attention is pre-trained (see Appendix - Section A) on CLEVR's scene graph annotations to compute an object-centric representation of an image, which is a tensor of unnormalised probabilities $\mathbf{Z} \in \mathbf{R}^{B \times O \times D}$, where B is the batch size, $O = 10$ is the maximum amount of objects identified in the image, and $D = 19$ corresponds to the number of attributes that describe each object. These attributes can fall within 5 categories, indexed as follows: $D_{0:3} =$ 3D coordinates and presence of the object, $D_{4:6} = 3$ possible shapes (sphere, cube, cylinder), $D_{7:8} = 2$ sizes (large, small), $D_{9:10} = 2$ materials (rubber, metal) and $D_{11:18} = 8$ colours (cyan, blue, yellow, purple, red, green, grey and brown). A single slot represents an object identified in the image, and the associated 19-dimensional vector refers to an unnormalised probability distribution over the attributes of this object. The output of the Slot Attention is binarised by taking the argmax of the value of each category, ignoring the first 4 indices corresponding to the position and presence of the object. This binarised $\mathbf{Z} \in \mathbf{R}^{B \times O \times D}$ tensor is processed by the Set Transformer, the reasoning component. This Set Transformer is an attention-based, encoder-decoder neural network which maps a set of vectors to a single vector, assuming permutation symmetry in the input (Lee et al., 2019). In the context of solving CLEVR-Hans3, the input vectors refer to the output of the slot by the Slot Attention, which is then mapped to an output vector representing the probability distribution over the 3 classes. Specifically, during training, the Set Transformer learns through its attention mechanism which attributes of the slots are the most relevant when deciding what is the most likely class label for an image. This process of sorting out which attributes to attend to in order to derive the class label is what is referred to as "reasoning" in their paper.[1] Therefore, the assembly of this neural Slot Attention and reasoning Set Transformer is referred to as a NeSy architecture, inheriting the inductive biases and functional properties present in both components, allowing efficient cooperation for solving CLEVR-Hans3. These inductive biases refer to object localisation and object attribute decomposition by the Slot Attention, along with classification task-driven attribute selection by the Set Transformer.

Altogether, this hybrid architecture consists of around 540K parameters, out of which only 158K belonging to the Set Transformer are tunable during training. The Set Transformer learns which attributes of the slots are the most relevant to reason over in order to classify the image. The remaining parameters from the Slot Attention are fixed (see Table 1).

We mainly build upon Stammer et al. (2021)'s work on contrasting the performances of a canonical CNN architecture, a ResNet pre-trained on ImageNet, with that of the NeSy model. Both are trained, validated and tested on CLEVR-Hans3 with real-life complications imputed. Additionally, in order to build more comprehensive experiments with respect to Stammer et al. (2021), we tested 2 new model variants which serve as intermediate models within the spectrum from pure deep learning to full neuro-symbolic modelling. This is because in their original work, we note unfairness in model comparison given that their ResNet34, whilst also pre-trained like the Slot Attention and having 40 times as many parameters, looked at an essentially different dataset, ImageNet. Its NeSy counterpart, on the other hand, was trained on CLEVR's scene annotations. From this comparison also stems a series of questions such as whether the inductive biases of the NeSy's Slot Attention or Set Transformer are irreplaceable, or whether either component can be substituted with fully-connected (FC) convolutional layers which are not designed for object attribute decomposition (perception) or attribute selection (reasoning) when solving CLEVR-Hans3. In order to fully understand *what* is the module that offers the NeSy architecture a good performance over CLEVR-Hans3 with real-life complications with respect to its pure deep learning counterpart, we thereby consider the following new model variants:

- **Perceptual CNN (FC layers + Set Transformer):** We propose a ResNet acting as the perception module coupled with a Set Transformer which would help us identify whether the FC convolutional network can replace the Slot Attention in learning how to provide an object-centric representation of an input image to the reasoning component in order to derive the correct class label.
- **Reasoning CNN (Slot Attention + FC layers):** We also propose a Slot Attention coupled with a ResNet acting as the reasoning component. This would help us sort out whether, despite lacking the inductive bias of the attention mechanism, the FC convolutional network can learn how to reason over the Slot Attention's object-centric representation.

These model variants would help us smooth the leap from pure deep learning to full NeSy modelling, enabling a more comprehensive, controlled comparison between both. Our results would also help us identify which is the dominant module (neural vs. reasoning), or whether the combination of both is responsible for robustly solving CLEVR-Hans3.

3. Methodology

We now proceed to outline how we designed the experimental conditions by modifying the CLEVR-Hans3 dataset, as well as how we assembled and tested the four model variants. Regarding the dataset with complications, we note that our base experimental condition is the original CLEVR-Hans3, which by default tests for generalisation under a visual confounder (colour). The remaining experimental conditions (noise, class imbalance, small data and mislabels) are independent of one another, but are built upon the base condition. We also only modify the training partition, whilst leaving the validation and test set unchanged. All images are down-scaled to visual dimensions 128 x 128 and normalised to have pixel values between -1 and 1 when fed to the model variants.

3.1. Complication Simulation

1. **Noise:** We add noise to the training partition of CLEVR-Hans3 by randomly selecting a third of the images from each class and subjecting them to Gaussian multiplicative noise. This consists of multiplying all the pixels of each image by a value sampled from a uniform distribution between 0 and 1 (see Figure 3).
2. **Class 1 Skewness:** We simulate this complication by creating a training dataset which includes 1/3 of the images from class 1 (which is the class containing the visual confounder). Therefore, the new dataset totalled 7000 images. The class 1 images to be included were selected randomly.
3. **Overall Small Data:** We create a new dataset by randomly selecting 1/3 of the images for each class, decreasing the training data size to 3000 samples.

[1] While we are aware of the contentious use of the word "reasoning" in the deep learning community, we note that terminology disagreement is not the focus of our work.

4. **Mislabels:** We randomly select 1/3 of the images of each class, and change the labels systematically by adding 1 and taking modulo 3. For example, the 1000 images originally belonging to class 2 will now be mislabeled as class 0.

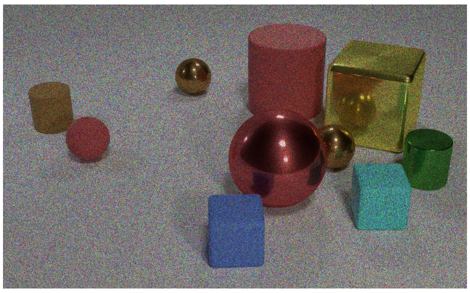

Figure 3. Sample of a class 0 CLEVR-Hans3 image with multiplicative noise added.

3.2. Model Assembly

We assembled our model variants as follows:

1. **Pure ResNet18:** We employed ResNet18 as our baseline. We opted for this smaller version in contrast to Stammer et al. (2021)'s ResNet34 for computational convenience during training. We load it pre-trained on ImageNet and cut off its output layer of 1000 nodes to replace it with a linear one over 3 classes before training it on CLEVR-Hans3. We note that despite having half the size of ResNet34, we are still able to closely replicate Stammer et al. (2021)'s results as Table 2 shows.

2. **Perceptual ResNet18 (FC layers + Set Transformer):** We employ ResNet-18 pre-trained on ImageNet as our FC layers to allow for a more controlled comparison amongst variants. This time, to couple the ResNet18 with the reasoning module (Set Transformer), we resized its last layer to output a $O \times D$ matrix, which is the same dimension as that given by the Slot Attention. All parameters are differentiable. Because ResNet18 is attempting to replace the Slot Attention as the perception module, we refer to this model variant as Perceptual ResNet18.

3. **Reasoning ResNet18 (Slot Attention + FC layers):** We assemble the Slot Attention pre-trained on CLEVR and a ResNet18 pre-trained on ImageNet for our FC layers, again aiming for a more controlled comparison. To link them together, the object-centric representation output by the Slot Attention is given as an input to the ResNet18. Here, the ResNet18 replaces the Set Transformer and acts as the "reasoning" module. Because the Slot Attention output is tridimensional $B \times O \times D$, lacking the dimension for channels, we concatenate three replicas of this object-centric representation to obtain $B \times 3 \times O \times D$ matrix that can be fed to the ResNet18. Following the specifications of Stammer et al. (2021), we keep the parameters of the Slot Attention fixed in this variant (Table 1).

4. **Concept Learner (Slot Attention + Set Transformer):** Our variant for full NeSy modelling would be the same one described in Section 2

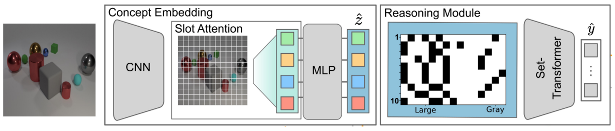

Figure 4. NeSy Pipeline. The Slot Attention works by encoding an input image into a tensor which is an object-centric representation of it. For example, given an image, it would output a matrix (or tensor where the batch size is 1) of size $O \times D$. We set $O = 10$ as that is the maximum amount of objects appearing in a scene, and $D = 19$ represents the task-dependent attributes. These slots are permutation symmetrical, i.e., it does not assume that objects identified in an image appear in any particular order. The Set Transformer works by taking this set of vectors and mapping it to a probability distribution of being assigned a particular class. The original NeSy model also had a visual and semantic explainer which were ignored in this paper out of concerns for fairness in model comparison, e.g., our model variant ResNet + Set Transformer is unable to produce sound saliency maps since ResNet is unable to contribute with a disentangled representation of a CLEVR-Hans3 scene unlike the Slot Attention. Figure cropped from (Stammer et al., 2021).

4. Experiments and Discussion

According to our motivation presented in Section 2, we designed a series of comprehensive experiments to explore two questions. The first one is whether the Concept Learner architecture is still more robust than pure ResNet architecture when facing real-life complications on training data. If yes, our second question will be whether the slot attention, set transformer or the combination of both plays a more important role.

Our implementations followed and extended the work of Stammer et al. 2021.[2] Since we have five training datasets with different real-life complications (Section 2.1) and four model architectures (Section 3), there are twenty (4×5) experiments in total. We trained them by minimizing the cross-entropy loss with the same hyperparameter configuration: 50 training epochs, a learning rate of $1e^{-4}$, the batch size of 128,[3] Adam optimizer $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8}$ and zero weight decay. Models including the Slot Attention have three more hyperparameters set: 10 slots, 3 iterations per slot attention, and 19 attributes. For variants where either the Slot Attention or Set Transformer is used, the

[2] Stammer et al. 2021's repository: NeSyXIL.

[3] We mostly recreated Stammer et al. (2021)'s hyperparameter setting, except that they train their ResNet34 for 100 epochs with a batch size of 64

learning rate decays as per a cosine annealing scheduler until a minimum of $1e^{-6}$. Note that for the Reasoning ResNet18 and the Concept Learner models, only a portion of the parameters is trained (Table 1). In addition, to quantify the uncertainty of our models' performances, each experimental condition for each model was run five times using random seeds: 0, 5, 25, 42 and 88. Then we used the same validation (confounded by colour) and test datasets (non-confounded) to evaluate each model's performance.

Table 1 shows the information of each model architecture, highlighting the parsimony of the Concept Learner since it has exponentially fewer parameters than its counterparts.[4] Each complication condition is built upon the original CLEVR-Hans3, which is confounded by colour. Therefore, if a model is not trained on the base condition, we assume the upper bound of its performance will be the corresponding classification accuracy achieved on the base condition. Before running our experiments, we first reproduced the results from Stammer et al. 2021, comparing a ResNet model and Concept Learner trained and tested on the confounder condition. Table 2 shows our reproduced results, which indicates that we achieved similar performance with our ResNet18 and nearly the same performance for the Concept Learner. This reproduction validates our subsequent experiments.

Model Architecture			Name	Number of Parameters	
Slot Attention	ResNet18	Set Transformer		Overall	Trainable
	✓		Pure ResNet18	11.2M	11.2M
	✓	✓	Perceptual ResNet18	11.4M	11.4M
✓	✓		Reasoning ResNet18	11.6M	11.2M
✓		✓	Concept Learner	539K	158K

Table 1. Information on model architectures, including alias and the number of parameters. We highlight that the full NeSy architecture is the most parsimonious in terms of model size.

	Pure ResNet		Concept Learner	
	Val Acc	Test Acc	Val Acc	Test Acc
Stammer et al. 2021	0.996±0.001	0.703±0.003	0.986±0.003	0.817±0.031
Reproduction	0.971±0.003	0.662±0.004	0.980±0.006	0.804±0.026

Table 2. Validation and test accuracy comparison between the results provided by Stammer et al. 2021 and our reproduction. Stammer et al. 2021 uses ResNet34 while our reproduction uses ResNet18 for the purpose of computational convenience.

Figure 5 shows all our experiment results. For each complication condition, the model that achieves the highest test accuracy is considered the most robust model w.r.t that condition. The full results can be found in Table 3. Across data complication conditions, the test accuracy of Reasoning ResNet18 and Concept Learner is always higher than Pure ResNet18 and Perceptual ResNet18. This shows that architectures employing Slot Attention always outperform architectures lacking it. This is mainly owed to the Slot Attention's ability to output an object-centric representation that is informative for the reasoning component to identify the correct label. In contrast, the Perceptual ResNet generally performs poorly because the reasoning module can not learn to pick the right attributes since the ResNet does not have inductive biases for object localisation and object attribute decomposition. In other words, the ResNet's output representation is not interpreted effectively by the Set Transformer. In addition, as Table 3 shows, the baseline (pure ResNet18) nearly always shows ceiling performance on training data, but systematically displays the worst test accuracy, meaning it is most affected by the visual confounder and the different data complications.

Our results show that the Reasoning ResNet18 is the most robust architecture for confounder, noisy data and mislabel complications, while the Concept Learner is the most robust architecture for class imbalance and small training data complications. We proceed to dive into discussing each:

4.1. Noise

The results obtained from the noisy data complication closely match that of the base condition, accounting for variance. Our results in this condition only suggest that slight perturbations of the training data do not affect each model's performance when being evaluated on clean testing data. In future work, we can explore adding more disruptive noise in order to observe different results. This could be in terms of sampling noise values from a high-variance distribution or imputing adversarial noise attacks.

4.2. Class 1 Skewness

We observe a drop in the performance of architectures lacking the Slot Attention. We argue that this indicates that these models can not disentangle the attributes of the objects present in the training image, thus making them vulnerable to class skewness, since there are fewer sample images from which they can generalise. This highlights how the Slot Transformer's ability to output an object-centric representation is helpful in building a robust classifier in case of facing an imbalanced training distribution. We argue that this object-centric representation facilitates the training of the reasoning component of the model as the latter only has to identify which attributes are the most relevant when deciding a class label, thus tolerating less amount of data for a particular class. The Concept Learner only performs marginally better than the Reasoning Resnet18, suggesting that the ResNet can perhaps reason over the attributes output by the Slot Attention. However, if we account for the number of parameters, the Concept Learner's higher performance, although marginal, highlights one of the strengths of NeSy AI: being parsimonious and not data-hungry.

4.3. Small Training Data

Our results on holding out an overall 2/3 of the training data yield performance measures closely resembling the patterns of the results to Section 4.2. It is not surprising that both the Pure ResNet18 and Perceptual ResNet18 perform worse than their base condition counterparts as they have seen fewer data during training. Consequently, they are even less able to disentangle the confounder and generalise robustly when compared with their counterparts which have

[4] All parameters of ResNet with/without Set Transformer are trainable, while parameters of Slot Attention can not be trained.

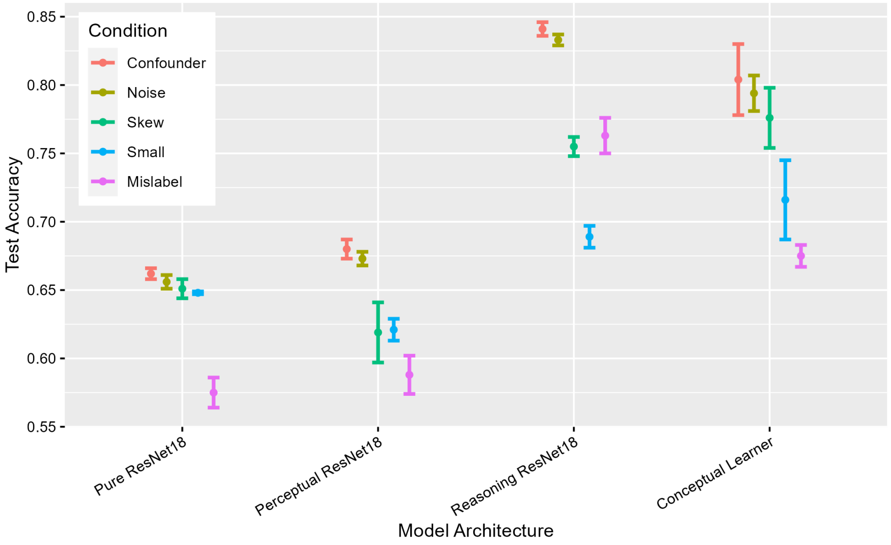

Figure 5. Visualisation of test accuracy for four model architectures (Section 3) building on five training datasets (Section 2.1), each model is trained five times with different seeds on 5 complication conditions, where confounder is the basis for the rest of complications. Points represent the mean test accuracy and the bars represent their standard deviations.

Model Architecture	Confounder (Base Condition)			Noise (1/3)			Skew (1/3 of Class 1)			Small (1/3)			Mislabel (1/3)		
	Train Acc	Val Acc	Test Acc	Train Acc	Val Acc	Test Acc	Train Acc	Val Acc	Test Acc	Train Acc	Val Acc	Test Acc	Train Acc	Val Acc	Test Acc
Pure ResNet18	1±0	0.971±0.003	0.662±0.004	1±0	0.968±0.003	0.656±0.005	1±0	0.954±0.004	0.651±0.007	1±0	0.947±0.013	0.648±0.01	1±0	0.819±0.03	0.575±0.011
Perceptual ResNet1	1±0	0.968±0.005	0.68±0.007	1±0	0.96±0.004	0.673±0.005	0.957±0.055	0.894±0.043	0.619±0.022	0.986±0.015	0.868±0.007	0.621±0.008	0.809±0.067	0.833±0.017	0.588±0.014
Reasoning ResNet1	0.928±0.003	0.925±0.005	**0.841±0.005**	0.925±0.003	0.927±0.007	**0.833±0.004**	0.882±0.005	0.865±0.002	0.775±0.007	0.807±0.004	0.79±0.008	0.689±0.008	0.733±0.004	0.873±0.007	**0.763±0.013**
Concept Learner	0.984±0.004	0.98±0.006	0.804±0.026	0.983±0.002	0.98±0.005	0.794±0.013	0.977±0.005	0.973±0.004	**0.776±0.022**	0.968±0.005	0.962±0.006	**0.716±0.029**	0.784±0.005	0.958±0.003	0.675±0.008

Table 3. Experiments results for four model architectures (Section 3) building on five training complications datasets (Section 2.1), where bold results represent the highest test accuracy (i.e., the most robust architecture) on certain real-life data complications.

the Slot Attention. Due to the Pure ResNet18 failing to act as a replacement for the Slot Attention in decomposing the image into an object-centric representation, the Perceptual ResNet18 can be thought of as simply a larger ResNet18. Despite having all parameters differentiable, the ResNet18 only has the inductive bias of handling shift and translational invariance in the input image (Uang et al., 1994). It lacks the ability of object-localisation amongst multiple co-appearing objects and attribute decomposition like the Slot Attention. Therefore, its parameters can not be adjusted in a way that helps it build better object-centric representations.

The best overall performance on the test set is achieved by the Concept Learner, albeit marginally better compared to the Reasoning ResNet18. This is consistent with previous findings by (Mao et al., 2019) where their NeSy model could achieve state-of-the-art performance on CLEVR despite holding out 90% of the training data.

4.4. Mislabels

The results obtained in this complication condition are perhaps the most interesting to discuss as they allow us to substantiate the claims that the Set Transformer is performing reasoning over the Slot Attention's object-centric representation, while the Reasoning ResNet18 is only identifying statistical correlations between the Slot Attention's attributes and labels.

Firstly, we observe the largest drop in performance for variants which do not employ the Slot Attention, with respect to their corresponding base experimental condition. It is no surprise that in addition to the weaknesses of these 2 variants discussed previously, for each class there were 1/3 of the images which contained features inconsistent with the labelling guideline of CLEVR-Hans3, which could have further confused both variants.

Secondly, the mislabel complication condition is also one where the Concept Learner performs significantly worse than the Reasoning ResNet18 on the test set. Higher training accuracy as shown in Table 3 in this condition however suggests that the Concept Learner has managed to reason over the attributes. Nonetheless, because 1/3 of the images per class had been mislabelled, its reasoning component has perhaps learned that class assignments can overlap. For example, the Set Transformer may have confused that class 1 images could correspond to images with EITHER "a small metal cube and small (metal) sphere" according to 2000 images OR "a large blue sphere and small yellow sphere" according to 1000 mislabeled images. In this particular case, when the Concept Learner is evaluated on the test set it performs poorly because there are no mislabels and class assignment is mutually exclusive, so it is likely to misassign a test image believing it is fulfilling the EITHER OR condition. We argue that this most likely sheds light on why the Reasoning ResNet18 can perform better than the Concept Learner in the test set: Reasoning ResNet18 has probably only captured the correlation between the most common features, facilitated by the Slot Attention, with their most common labels, as denoted by 2/3 of the correctly labelled images. Because "reasoning over attributes" is not just a statistical matter of capturing the most frequent co-occurrences between input features and output labels, we argue that the mislabel complication condition serves as evidence that the Reasoning ResNet18 is not performing the same kind of reasoning the Concept Learner does. Rather, the Reasoning ResNet18 is matching frequent attributes to corresponding frequent class labels. This argument also applies to the Class Skewness and Small Training Data complications, explaining why the Reasoning ResNet18's training accuracies are significantly lower compared to that of the Concept Learner (see Table 3): for this architecture there is just not enough sample images of either one class or all classes to associate the most common label. In contrast, for the Concept Learner, learning which images are assigned to a particular class is not a matter of statistical correlation, but rather identifying which attributes of the Slot Attention it should pay attention to derive the class label.

Furthermore, performing well on this complication condition is not necessarily a good sign because the systematically placed mislabels make the training data images easily ambiguous. Thus, a reasoner which performs poorly could be understood as showing that it is "paying attention" to the right attributes.

4.5. Overall

Given the above results, our research questions can be answered as follows. For all real-life training data complications proposed, the Concept Learner architecture is always more robust than Pure ResNet18 architecture. When comparing the importance of Slot Attention and Set Transformer, the former is essential for object-centric representation while the latter can be replaced by FC layers (e.g., ResNet18). However, an important take-away from these experiments is that ResNet18 is less parsimonious than the Set Transformer as a 'reasoning module', and this is despite it being pre-trained on ImageNet, thus greatly outweighing the Set Transformer in terms of the amount of supervision and size. Furthermore, it is open to debate whether FC layers can perform reasoning as the Set Transformer does, rather than just identifying statistical correlations, as evidenced by the mislabel condition. Considering the number of parameters, computational efficiency or dataset size, then the Slot Attention coupled + Set Transformer architecture is arguably more suitable for this task. Besides, another takeaway of our experiments is that NeSy AI is particularly attractive in how it decomposes a problem in order to solve it. In the CLEVR-Hans3 classification task, it first tries to find a disentangled representation of an image and then assigns a class label based on this representation. This is in contrast with the traditional approach taken by pure deep learning which seeks to map an input image directly to a label, leaving the intermediate representation as a black-box. We thus argue that inductive biases such as object localisation and object attribute decomposition from the Slot Attention, coupled with attribute selection through attention by the Set Transformer, complement each other harmoniously and efficiently in a hybrid architecture in order to robustly solve this task, compared to alternative approaches.

5. Conclusions

One of the main traits of NeSy AI is the compilation of prior knowledge into neural architectures to constrain learning, helping build robust classifiers. For example, Slot Attention's inductive bias of object localisation and attribute decomposition is useful in obtaining an informative representation of the input space for a subsequent reasoner to process it and solve a task. Such integration helps build more parsimonious models which are not data-hungry, and are robust to possible complications appearing in the training distribution.

Our results corroborate the growing literature exploring the strengths of NeSy modelling. Specifically, the experiments we run show that NeSy modelling helps build parsimonious classifiers that are robust to noisy, heavily skewed and small training data. Our NeSy model, however, is vulnerable to mislabels in the dataset, which we argue is not necessarily a drawback, because it exposes possible pathological features of a training dataset.

Additionally, we recall that one of the main motivation behind our work is to assess whether NeSy AI can tolerate data complications inspired by real-life datasets like healthcare ones. This is partly due to the research on NeSy AI mainly limited to laboratory conditions. Our results drive us to believe that NeSy AI holds great, untapped potential for solving real-life tasks such as disaster or disease classification, where it can highlight its strengths of parsimonious modelling through constrained learning, and incorporating human prior knowledge. Another interesting future direction is to explore NeSy architectures which do not assume a pre-trained Slot Attention, but rather train it from scratch with backpropagated training signal from the reasoning module. Training DeepProbLog (Manhaeve et al., 2018) to solve CLEVR-Hans3 represents one such example. There, the user only has to specify a probabilistic logic program and assemble neural networks which learn to predict the probabilities of the logic predicates in the program. The networks do not require to be pre-trained using CLEVR scene graph annotations.

References

Bahri, Yasaman, Dyer, Ethan, Kaplan, Jared, Lee, Jaehoon, and Sharma, Utkarsh. Explaining neural scaling laws. *arxiv*, 2021. URL https://arxiv.org/pdf/2102.06701.pdf.

Beede, Emma, Baylor, Elizabeth, Hersch, Fred, Iurchenko, Anna, Wilcox, Lauren, Ruamviboonsuk, Paisan, and Vardoulakis, Laura M. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–12, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376718. URL https://doi.org/10.1145/3313831.3376718.

Biderman, Stella and Scheirer, Walter J. Pitfalls in machine learning research: Reexamining the development cycle. In Zosa Forde, Jessica, Ruiz, Francisco, Pradier, Melanie F., and Schein, Aaron (eds.), *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, pp. 106–117. PMLR, 12 Dec 2020. URL https://proceedings.mlr.press/v137/biderman20a.html.

Brigato, Lorenzo and Iocchi, Luca. A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2490–2497, 2021. doi: 10.1109/ICPR48806.2021.9412492.

Carion, Nicolas, Massa, Francisco, Synnaeve, Gabriel, Usunier, Nicolas, Kirillov, Alexander, and Zagoruyko, Sergey. End-to-end object detection with transformers. *Computer Vision – ECCV 2020*, pp. 213–229, 2020. doi: 10.1007/978-3-030-58452-8_13.

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

Garcez, Artur d'Avila and Lamb, Luis C. Neurosymbolic ai: the 3rd wave. *arXiv preprint arXiv:2012.05876*, 2020.

Johnson, Justin, Hariharan, Bharath, van der Maaten, Laurens, Fei-Fei, Li, Zitnick, C. Lawrence, and Girshick, Ross. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 07 2017. doi: 10.1109/cvpr.2017.215.

Kaplan, Jared, McCandlish, Sam, Henighan, Tom, Brown, Tom B., Chess, Benjamin, Child, Rewon, Gray, Scott, Radford, Alec, Wu, Jeffrey, and Amodei, Dario. Scaling laws for neural language models. *arXiv:2001.08361 [cs, stat]*, 01 2020. URL https://arxiv.org/abs/2001.08361.

Karimi, Davood, Dou, Haoran, Warfield, Simon K., and Gholipour, Ali. Deep learning with noisy labels: exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, pp. 101759, 06 2020. doi: 10.1016/j.media.2020.101759.

Kucker, Sarah C, Samuelson, Larissa K, Perry, Lynn K, Yoshida, Hanako, Colunga, Eliana, Lorenz, Megan G, and Smith, Linda B. Reproducibility and a unifying explanation: Lessons from the shape bias. *Infant Behavior and Development*, 54:156–165, 2019.

Lee, Juho, Lee, Yoonho, Kim, Jungtaek, Kosiorek, Adam, Choi, Seungjin, and Teh, Yee Whye. Set transformer: A framework for attention-based permutation-invariant neural networks. In Chaudhuri, Kamalika and Salakhutdinov, Ruslan (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Locatello, Francesco, Weissenborn, Dirk, Unterthiner, Thomas, Mahendran, Aravindh, Heigold, Georg, Uszkoreit, Jakob, Dosovitskiy, Alexey, and Kipf, Thomas. Object-centric learning with slot attention. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Manhaeve, Robin, Dumancic, Sebastijan, Kimmig, Angelika, Demeester, Thomas, and De Raedt, Luc. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

Mao, Jiayuan, Gan, Chuang, Deepmind, Pushmeet, Tenenbaum, Joshua, and Wu, Jiajun. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arxiv*, 04 2019. URL https://arxiv.org/pdf/1904.12584.pdf.

Marcus, Gary. The next decade in ai: Four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

Miotto, Riccardo, Wang, Fei, Wang, Shuang, Jiang, Xiaoqian, and Dudley, Joel T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19:1236–1246, 2018. doi: 10.1093/bib/bbx044. URL https://www.ncbi.nlm.nih.gov/pubmed/28481991.

Redman, Thomas. If your data is bad, your machine learning tools are useless, 04 2018. URL https://hbr.org/2018/04/if-your-data-is-bad-your-machine-learning-tools-are-useless.

Stammer, Wolfgang, Schramowski, Patrick, and Kersting, Kristian. Right for the right concept: Revising neurosymbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3619–3629, 2021.

Sánchez Fernández, Iván, Yang, Edward, Calvachi, Paola, Amengual-Gual, Marta, Wu, Joyce Y., Krueger, Darcy, Northrup, Hope, Bebin, Martina E., Sahin, Mustafa, Yu, Kun-Hsing, and Peters, Jurriaan M. Deep learning in rare disease. detection of tubers in tuberous sclerosis complex. *PLOS ONE*, 15:e0232376, 04 2020. doi: 10.1371/journal.pone.0232376.

Theodoridis, Sergios. *Chapter 18.1.1 - Distributed Representation*, pp. 953. Machine Learning (Second edition) A Bayesian and Optimization Perspective. Academic Press, 2020. URL https://www.sciencedirect.com/science/article/pii/B9780128188033000301#s0155.

Tuli, Shikhar, Dasgupta, Ishita, Grant, Erin, and Griffiths, Thomas L. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021.

Uang, Chii-Maw, Yin, Shizhuo, Andres, P., Reeser, Wade, and Yu, Francis T. S. Shift-invariant interpattern association neural network. *Appl. Opt.*, 33(11):2147–2151, Apr 1994. doi: 10.1364/AO.33.002147. URL https://opg.optica.org/ao/abstract.cfm?URI=ao-33-11-2147.

Wang, Xuezhi, Wang, Haohan, and Yang, Diyi. Measure and improve robustness in nlp models: A survey. *arXiv preprint arXiv:2112.08313*, 2021.

Xia, Gui-Song, Hu, Feng, Zhang, Liangpei, and Lu, Xiaoqiang. A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

Zhou, Yuxi, Hong, Shenda, Shang, Junyuan, Wu, Meng, Wang, Qingyun, Li, Hongyan, and Xie, Junqing. Addressing noise and skewness in interpretable health-condition assessment by learning model confidence. *Sensors*, 20:7307, 12 2020. doi: 10.3390/s20247307.

A. Pre-training of the Slot Attention

The Slot Attention (see Locatello et al., 2020) works by first encoding an input image into N input features using convolutional layers augmented with positional embeddings. These input features are mapped to a set of K slots, each slot being a vector of probabilities of dimension D. The objective of Slot Attention is to train its parameters so that slots learn to identify which objects are in an image and what are its attributes. During training, at each iteration, slots "compete" for explaining parts of the input via a softmax-based attention mechanism and update their representation using a recurrent update function. The goodness of the prediction of the Slot Attention is measured via the Hungarian Loss (see (Carion et al., 2020), which is traditionally used as an optimisation objective in object detection tasks regarding the prediction of coordinates of bounding boxes.

Stammer et al. (2021) pre-train the Slot Attention on CLEVR scene graph annotation with a cosine annealing learning rate scheduler for 2000 epochs, minimum learning rate of $1e^{-5}$, initial learning rate $4e^{-4}$, batch size of 512, 10 slots, 3 internal slot-attention iterations and the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8}$ and zero weight decay.