Can deep convolutional neural networks support relational reasoning in the same-different task?

Guillermo Puebla School of Psychological Science, University of Bristol, UK

Jeffrey S. Bowers School of Psychological Science, University of Bristol, UK

Same-different visual reasoning is a basic skill central to abstract combinatorial thought. This fact has lead neural networks researchers to test same-different classification on deep convolutional neural networks (DCNNs), which has resulted in a controversy regarding whether this skill is within the capacity of these models. However, most tests of same-different classification rely on testing on images that come from the same pixel-level distribution as the training images, yielding the results inconclusive. In this study, we tested relational same-different reasoning in DCNNs. In a series of simulations we show that models based on the ResNet architecture are capable of visual same-different classification, but only when the test images are similar to the training images at the pixel level. In contrast, when there is a shift in the testing distribution that does not change the relation between the objects in the image, the performance of DCNNs decreases substantially. This finding is true even when the DCNNs' training regime is expanded to include images taken from a wide range of different pixel-level distributions or when the model is trained on the testing distribution but on a different task in a multitask learning context. Furthermore, we show that the relation network, a deep learning architecture specifically designed to tackle visual relational reasoning problems, suffers the same kind of limitations. Overall, the results of this study suggest that learning same-different relations is beyond the scope of current DCNNs.

Introduction

Feedback connections, can be investigated, and methods relational reasoning is core to human intelligence (Penn et al., 2008) and has proven to be a challenge for an earlier generation of connectionist models (e.g., O'Reilly & Busby, 2001; Rogers & McClelland, 2004; St. John, 1992), as well as more recent deep neural networks (for recent reviews, see Ricci et al., 2021; Stabinger et al., 2021). Perhaps the simplest form of relational reasoning is the same-different task that simply requires the reasoner to determine whether two inputs are the same or different by some criterion. In the domain of vision, the simplest version of this is to classify images as visually identical or not. This skill, essential to abstract combinatorial thought, is much more developed in humans and chimpanzees than in other species (Gentner et al., 2021) and develops early in human infants (e.g., Ferry et al., 2015).

Recently, there has been mixed evidence regarding whether standard deep convolutional neural networks (DCNNs) can support same-different matching of images. Much of this research has used the synthetic visual reasoning test (SVRT) developed by Fleuret et al. (2011). This dataset comprises sets of 23 classification problems involving images, of randomly generated shapes (for example images, see Figure 1). In their study, Fleuret et al. (2011) found that the standard machine learning techniques of the time performed poorly, whereas most humans were able to solve the problems after seeing a few examples. Similarly, Stabinger et al. (2016) showed that state-of-the-art DCNNs (at the time) LeNet and GoogLeNet performed poorly on the same SVRT same-different tasks, and more recently, Kim et al. (2018) showed that vanilla DCNNs were poor at SVRT same-different tasks, and using a different dataset, showed that the Santoro et al. (2017) relational network also failed to support same-different judgments.

Interestingly, Kim et al. (2018) did find that a Siamese network (Bromley et al., 1993) that encoded the two shapes in two separate channels to simulate the effects of attentional selection and perceptual grouping, learned to classify images as "same" or "different" easily, leading the authors to conclude that object individuation is a key step in solving the same-different task. At the same time, they also argue that a full solution to the same-different problem requires a network to encode dynamic representations of relations rather than statically storing visual relation templates in synaptic weights. That is, in their view, symbolic processes need to be implemented to fully solve the same-different task.

Citation: Puebla, G., & Bowers, J. S. (2022). Can deep convolutional neural networks support relational reasoning in the same-different task?. *Journal of Vision*, *22*(10):11, 1–18, https://doi.org/10.1167/jov.22.10.11.

https://doi.org/10.1167/jov.22.10.11 Received November 26, 2021; published September 12, 2022 ISSN 1534-7362

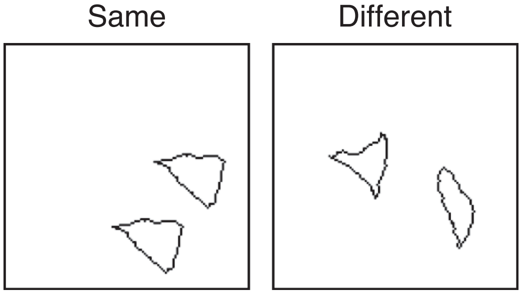


Figure 1. Examples of the “same” and “different” categories from SVRT problem #1. In this problem, an image belongs to the category same if both shapes are identical up to translation on the canvas and different otherwise.

In contrast, there are recent reports that the current state-of-the-art DCNNs can learn the same-different task. If this is indeed the case, it would be a striking example of standard networks solving a fundamental relational reasoning task without implementing any symbolic machinery. Funke et al. (2021) noted that Kim et al. (2018) only tested relatively small CNNs (up to six layers), and when they replicated the same-different experiments on the SVRT dataset using ResNet-50 (He et al., 2016), a network of 50 layers, the models were able to perform the task successfully. Funke et al. (2021) noted that the success does not necessarily imply DCNNs can perform well on all visual reasoning tasks, but they do highlight that standard feedforward processing DCNNs can learn the (in-distribution) same-different task and that Kim et al.’s claim regarding the need for extra mechanisms for abstract visual reasoning is unwarranted.

Similarly, Messina et al. (2021) have shown that a range of recent DCNNs, specifically ResNet, DenseNet (Huang et al., 2017), and CorNet-S (Kubilius et al., 2018), can solve the same-different SVRT tasks, whereas they confirm that this is difficult for older AlexNet (Krizhevsky et al., 2012) and VGG (Liu & Deng, 2015) networks. The authors conclude that “We think that the development of the abstract and relational abilities of neural networks is an important leap towards achieving some interesting new task.”

However, there is a fundamental problem with using success on the SVRT dataset as evidence that CNNs can support same-different relational reasoning. A key feature of relational reasoning is that it is reasoning based on relations between objects rather than any low-level visual details of the inputs. In the domain of visual reasoning, this entails that same-different discrimination should extend to novel images. The SVRT dataset does test models on novel images, but the test images are generated in the same way (i.e., the train and test datasets come from the same pixel-level distribution), and accordingly, it does not test the hypothesis that models have acquired the capacity to support relational reasoning on the same-different task.

Simulations

In the simulations described in this article, we test abstract same-different reasoning in several DCNNs models based on the ResNet-50 architecture. The basic tenet of our simulations is that a model that has learned the abstract *same* and *different* relations should be able to recognize examples of these relations beyond its training set. Similar to previous research on abstract visual reasoning (Funke et al., 2021, Study 1; Yan & Zhou, 2017), our approach used carefully constructed out-of-distribution samples to test whether DCNNs understood the trained concept or instead relied on superficial statistical cues present in the training data.

Our training and test data are based on problem #1 of the SVRT (see Figure 1). In this problem, images of two randomly generated shapes are classified as “same” if they are the same up to translation on the canvas and “different” otherwise. We created nine new datasets that followed the same abstract rule as problem #1 (see Figure 2). However, each new dataset was generated through a distinct stochastic generative process (i.e., a different pixel-level distribution). Each dataset was defined as follows:

- In the irregular dataset, each shape was a irregular polygon. These polygons were generated by sampling a series of 1 to 7 points of a circumference of radio π (uniformly sampled from 1 to 40 pixels) around a randomly chosen center. After this, we added uniformly distributed random noise to each point and connected all of them with straight lines. In the same category, both shapes were identical except for the position in the canvas. In the different category, the initial points and point errors of the second polygon were resampled such that both shapes were different.
- In the regular dataset, each shape was a regular polygon. These polygon were generated in the same way as the Irregular dataset except that we did not add with random noise to the polygon points.
- The open dataset was generated in the same way as the irregular dataset, except that the first and last vertices of each shape were not connected.
- The wider line dataset was generated in the same way as the irregular dataset, except that the line width was set to two pixels instead of one.
- The scrambled dataset was was generated in the same way as the Regular dataset except that in the different category, one the of the objects (scrambled) was generated by dividing the other object into sections and displacing them randomly around the center.
- The random color dataset was generated in the same way as the irregular dataset, except that for each image the line color was chosen randomly.

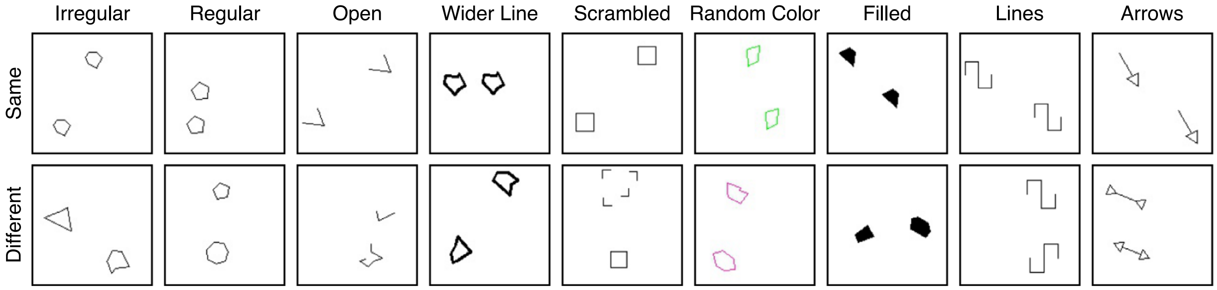


Figure 2. Examples of the same and different categories from our nine new versions of SVRT problem #1. See text for details.

- The filled dataset generated in the same way as the irregular dataset except that the shapes were filled with black.
- In the lines dataset, each object corresponded with a line created by joining two open squares, one with the opening pointing downward and the other with the opening pointing upward, at the end of the opposite left/right sides (i.e., the right side downward-facing open square was joined to the left side of the upward-facing square or vice versa); in the same category, the lines were identical, whereas in the different category, the second line was created by joining the open squares at the opposite left/right sides than the first line.
- In the arrows dataset, the objects were arrows consisting of one or two triangular head(s) and a line; the head(s) and the line were connected; in the same category the arrows were the same and in the different category the orientation of each head was inverted.

Note that, among these nine different stimulus sets there are differences in the level of low-level similarity with the original SVRT data. In particular, the irregular, regular, and, to a lesser extent, the open datasets are more similar to the original data than the rest of the datasets. The code to generate these datasets, as well as to run all our simulations, can be found at: https://github.com/GuillermoPuebla/same_different_paper.

Simulation 1: Generalization to unseen conditions

In simulation 1, we performed the most basic and stringent test of abstract relational reasoning. We trained several models on the original problem #1 and then presented with 5,600 images from each of the 10 stimulus test sets. That is, our testing conditions consisted of new images from the original training set (replicating Funke et al., 2021), and novel images from the other nine test datasets that were not seen during training. As noted elsewhere in this article, a model that has learned the abstract *same* and *different* relations should generalize learning on the same-different task independently from the pixel-level similarity to original SVRT data.

In simulation 1, we tested three sets of models based on the ResNet architecture. The first set consisted of four ResNet-50 classifiers. All models consisted of a ResNet-50 convolutional front end followed by a hidden layer with 1,024 units with ReLU activation (see Figure 3A). In simulation 1, there was one output layer consisting of a single sigmoid unit that predicted the probability that the input image belonged to the category same. We pretrained the models' convolutional front end using either ImageNet (Deng et al., 2009) or TU-Berlin (Eitz et al., 2012), a dataset of human-generated sketches. Furthermore, we varied how we treated the output of the convolutional front end before passing it to the hidden layer. We either applied a global average pooling (GAP) operation to the output, as in Funke et al. (2021), or flattened the output, as in Messina et al. (2021).[1]

The second set of models were different versions of the ResNet architecture that varied on depth. In particular we used ResNet-18, ResNet-34, ResNet-101, and ResNet-152 front ends with GAP pooling and ImageNet pretraining, because this was most successful condition in the first set of models. The goal of testing these models was to measure the potential role of network depth on the generalization of same-different discrimination.

The third set of models consisted of two variations of a relation network (Santoro et al., 2017). This architecture is especially relevant for the present study because it was explicitly designed to perform relational reasoning on the visual domain and it's fully compatible with DCNNs. As illustrated in Figure 3B, a relation network consists of a convolutional front end that outputs a series of filters and a relation module. The relation module organizes the filter activations into columns that correspond with specific positions across filters (denoted by different colors in Figure 3B), and generates all possible pairs of columns. All this pairs are processed by a single multilayer perceptron, g_θ, yielding a vector per pair. These vectors are summed up and passed trough a second multilayer perceptron, f_ϕ, that

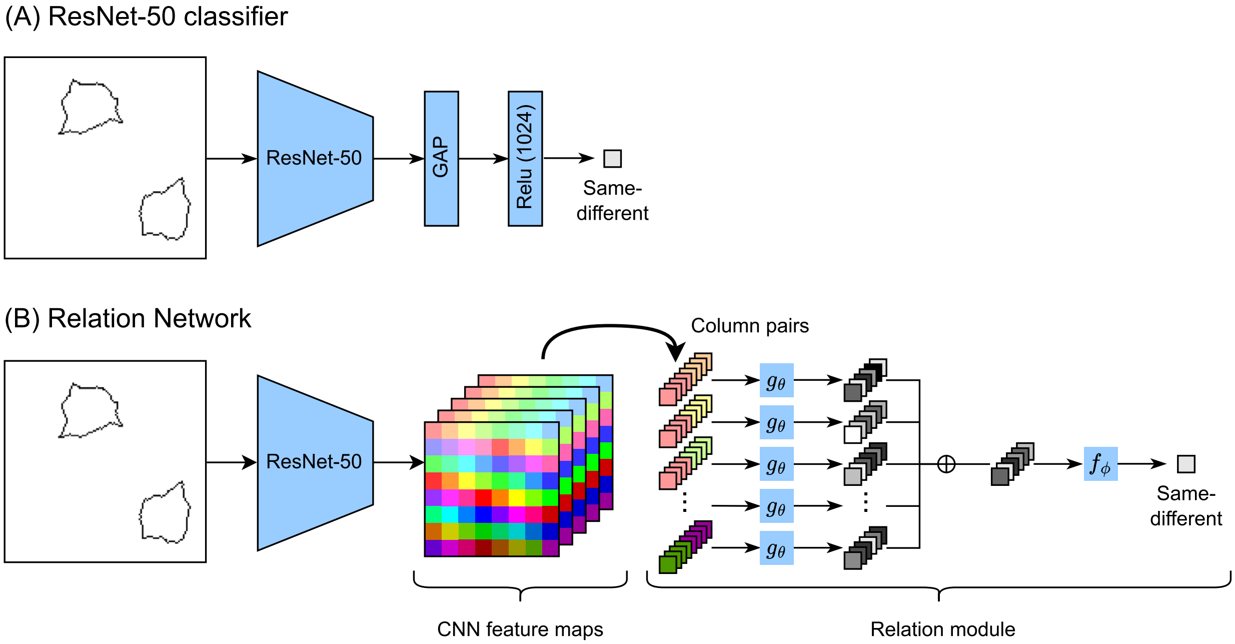


Figure 3. Models tested. (A) ResNet classifier. (B) Relation network.

yields the final same-different prediction. Note that the feature columns inputs to the relation module do not necessarily represent objects or objects parts. Instead, they represent whatever is in their corresponding receptive fields (e.g., the background, a texture, or even multiple objects at the same time). We created two versions of the relation network[2] by varying the filter inputs to the relation module. In the first version we used the output of the last convolutional layer of ResNet-50 (pretrained on ImageNet), that consisted of 2,048 4×4 filters. Because the original relation network of Santoro et al. (2017) used a CNN front end with filter outputs of size 8×8, in the second version we used the 1,024 output filters of the last convolutional layer of Resnet-50 with filter size 8×8.

Following the recommendations of Mehrer et al. (2020), who argue that network behavior should be based on groups of network instances, we trained 10 instances of each model. We used the Adam optimizer (Kingma & Ba, 2014). Training proceeded in two stages. In the first stage, the pretrained ResNet network was frozen while the rest of the network was trained with a learning rate of 0.0003. In the second stage, the complete model was trained with a learning rate of 0.0001. The training data consisted of the original data from SVRT problem #1. In the first stage, the model was trained on 28,000 images for 5 epochs with a batches of 64 samples. In the second stage, the model was trained on the same images for 10 epochs and with the same batch size.

Because same-different decisions were often performed on test datasets with different distributions than the training datasets, it is possible that there is a different optimal classification threshold for each test dataset. To account for this, we used the area under the receiver operating characteristic (ROC) curve (AUC), which is a performance measure that takes into consideration all possible classification thresholds. AUC values range from 0.0 to 1.0, where 0.5 corresponds with chance-level responding and 1.0 with perfect classification. The AUC can be interpreted as the probability that a randomly sampled example of the positive category (same) will be assigned a higher predicted probability than a randomly sampled example of the negative category (different) (Hanley & McNeil, 1982). We interpreted the AUC values according the general guidelines of (Hosmer et al., 2013) (Table 1).

AUC range	Category
[0.9−1.0]	Outstanding
[0.8−0.9)	Excellent
[0.7−0.8)	Acceptable
≤0.7	Poor

Table 1. AUC interpretation criteria.

Results and discussion

ResNet-50 classifiers: As can be seen in Figure 4, all models achieved outstanding performance in the original test dataset. Average validation AUC scores on the original dataset were greater than 0.9 for all models. Furthermore, we did not find large differences in generalization as a function of pooling or the pretraining dataset. Overall, the ImageNet and GAP

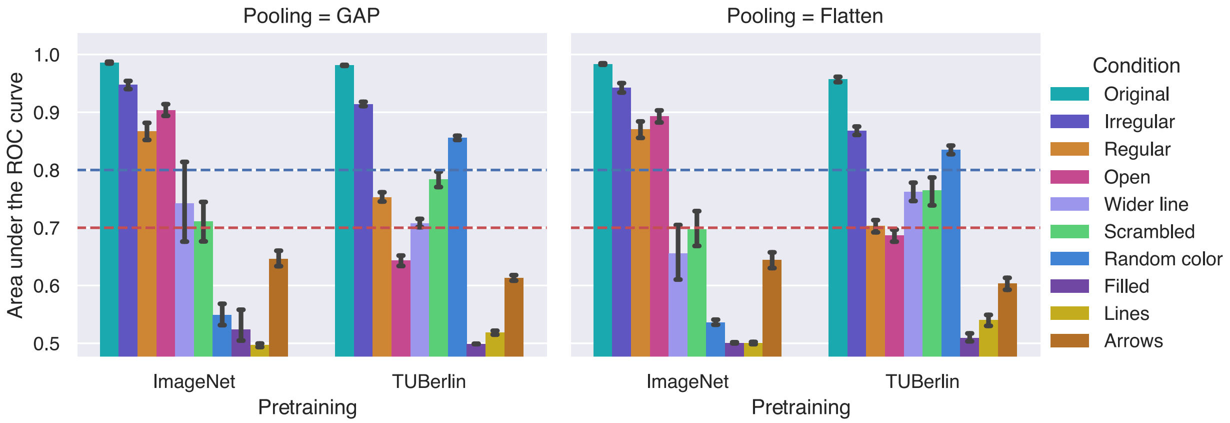


Figure 4. Mean AUC by ResNet-50 classifier across datasets in simulation 1. Error bars are 95% confidence intervals.

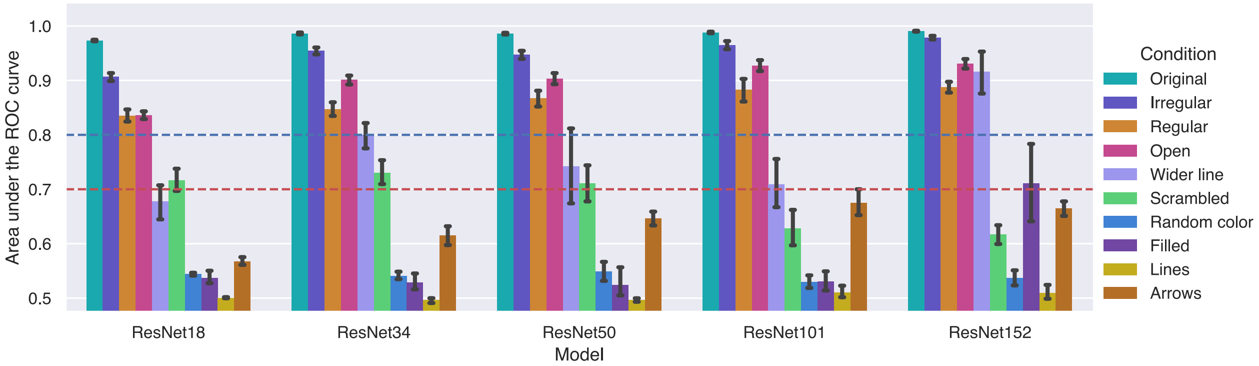


Figure 5. Mean AUC by ResNet version and test dataset on simulation 1. Error bars are 95% confidence intervals. Note: the ResNet-50 means correspond with the same data shown in Figure 4 (GAP and ImageNet).

model was the best performing model in the original test dataset, as well as, on average, in the nine new test datasets. Accordingly, the following analysis (as well as simulations 2 to 4) will concentrate on this condition. The ImageNet and GAP model showed outstanding or excellent performance in the irregular, regular, and open datasets. As can be appreciated in Figure 2, these datasets were the most featurally similar to the training data. In contrast, the ImageNet and GAP model showed a poor performance on the random color, filled, lines, and arrows datasets and a acceptable performance on the wider line and scrambled datasets. In general, these results show that generalization on the same-different task is very susceptible to distribution shifts that do not affect the relationship between the objects in the image. This pattern of results is inconsistent with the models learning the abstract *same* and *different* relations.

ResNet depth variations

As shown in Figure 5, all versions of ResNet performed similarly. Average validation AUC scores on the original dataset were greater than 0.9 for all models. Overall, the ResNet-152 version was the best performer, with significantly higher AUC on the wider line and filled conditions than the ResNet-50 model, although ResNet-152 achieved the worst performance in the scrambled condition. Given that we found some evidence for a role of network depth on generalization, we will include ResNet-50 and ResNet-152 models in all the following simulations. Importantly, all ResNet versions showed the same overall pattern of generalization, with better generalization to the conditions that were more similar to the training data, which is inconsistent with learning the abstract relations *same* and *different*.

Relation networks: Similar to the ResNet classifiers, both relational networks achieved outstanding performance in the original test dataset (Figure 6). Average validation AUC scores on the original dataset were above 0.9 for all models. Overall, the relation network with 8 × 8 filter inputs was the best performing model on the original test dataset as well as across the nine new test datasets (in many cases by a large margin). Accordingly, for the following analysis (as well as simulations 2 to 4), we will concentrate on it. This model achieved excellent performance or above in

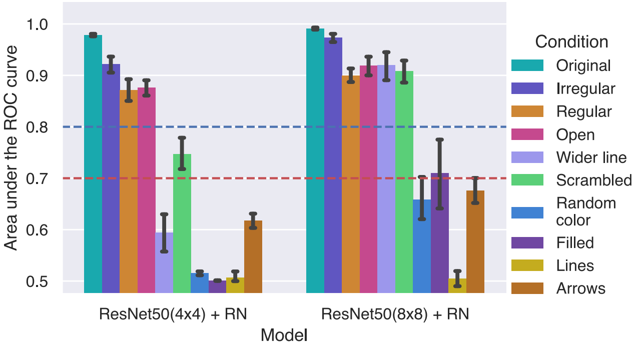


Figure 6. Mean AUC by relation network version and test dataset on simulation 1. Error bars are 95% confidence intervals.

the irregular, regular, open, wider line and scrambled datasets. Performance was acceptable on the filled dataset. Still, the relational network performed poorly on the random color, arrows, and Lines datasets, which is inconsistent with an understating of the abstract relations *same* and *different*.

Simulation 2: Leave-one-out training

One potential criticism to simulation 1 is that the training data (line drawings of random shapes) was not rich enough for the models to form a more complex representation of the same and different relations. Note that Messina et al. (2021) do interpret their results with the same training data as supporting relational same-different reasoning in DCNNs. Nevertheless, we agree that the representations of the human visual system are based on rich stimuli and, therefore, is important to test what happens when the models have access to a richer training set. Therefore, in simulations 2 to 4, we tested whether augmenting the training regime of the models would improve generalization on the same-different task to unseen stimuli. In simulation 2, we did this by using a leave-one-out procedure, where the models were trained on nine stimulus conditions consisting of images from the original SVRT data and all the new datasets except one (cf. Geirhos et al., 2018). For each condition, we trained 10 model instances with the same settings as in Simulation 1 except that the models were trained for 15 epochs instead of 10. We tested the models in the one stimulus set they were not trained on. For example, the models in the irregular stimulus condition were trained on the original data and all the new datasets except the irregular condition, in which they were tested on.

Results and discussion

Figure 17 in Appendix A presents all validation AUC scores for all trained datasets per model and condition. As can be seen in Figure 7, the relation network was the best performer with AUC excellent or above in all conditions, but lines and arrows. The ResNet-152 model performed worse than the ResNet-50 version, showing poor performance on the scrambled, random color, lines and arrows conditions. All models performed poorly in the lines condition. Furthermore, the performance of all models was near the lower limit for acceptable in the arrows condition. Overall, these results show that augmenting the training regime directly on the same-different task improved performance on untrained datasets for all models. However, this benefit does not seem to be based on a better understanding of the shared relational structure of the problem, given the results in the lines and arrows conditions.

Simulation 3: Multitask learning

In simulation 2, we augmented the models' experience by training on the same-different across a range of stimulus conditions. A potential criticism

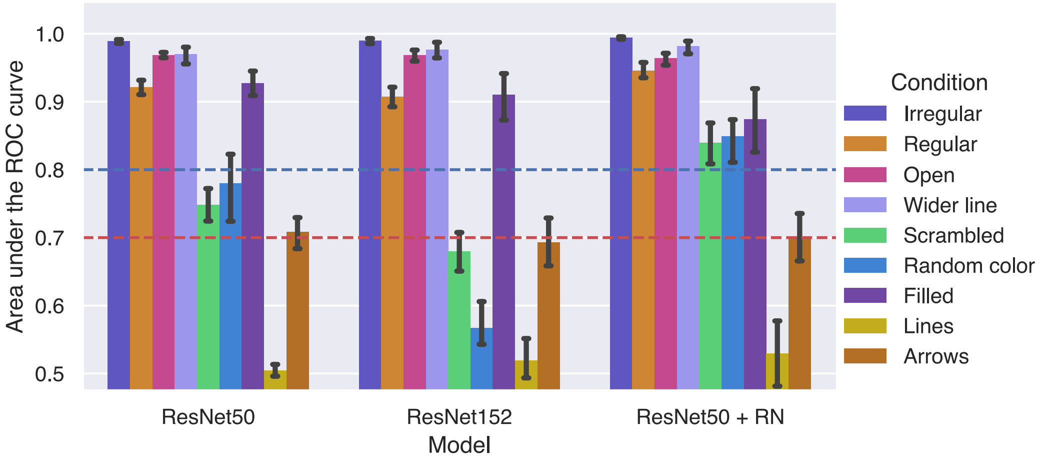


Figure 7. Mean AUC by test dataset and model on simulation 2. Error bars are 95% confidence intervals.

to this strategy is that it does not give the models any experience with the specific stimulus condition they are tested on. Accordingly, in simulation 3, we augmented the models' experience on all the datasets through multitask learning. In deep learning research, multitask learning has long been used as technique to improve generalization (for a review, see Ruder, 2017). In this simulation, the models were trained on two tasks. The first was the same-different task as in the previous simulations. In simulation 3a, the second task was a relative position task. This process consisted in classifying whether the lower object in the image was to the right of the upper object (category 1) or to the left (category 0). In simulation 3b, the second task consisted on classifying each same-different sample into its corresponding condition (i.e., original, irregular, regular, etc.). In both simulations, we trained both task jointly.

To train the ResNet models in simulation 3a, we added a second output layer with a single sigmoid unit. During training we presented the models with images from all conditions. However, we only allowed the models to learn to classify the images from the original condition as same or different, whereas the models learned to classify all images according to relative position. To this end, we used the following composed loss function:

$$\mathcal{L}_{\text{total}} = \sum_{i\in\text{batch}} w_i^{sd} \cdot CE(y_i^{sd}, \hat{y}_i^{sd}) + w_i^{rp} \cdot CE(y_i^{rp}, \hat{y}_i^{rp}), \quad (1)$$

where $CE(y, \hat{y})$ is the cross-entropy loss between the label y and the prediction $\hat{y}$, and w^{sd} and w^{rp} are the weights for the same-different loss and the relative position loss, respectively. During training, w^{rp} was set to 1 for all images. In contrast, when the model received images from the original SVRT data, we set w^{sd} to 1; otherwise, it was set to 0. During testing, we presented the models with images of each problem version and recorded the models' same-different and relative position AUC. All other training and testing parameters were the same as in simulation 1, except that we trained the models for 15 epochs rather than 10.

For the relation network, we added a question layer[3] and a second output layer with a single sigmoid unit. The relation network concatenates this question to all the column pairs, making the computation performed by g_θ question dependent. To train on the same-different task we created a vector, $[1\ 0]^T$, representing the same-different question and another vector, $[0\ 1]^T$, representing the relative position question. When the input to the question layer was the same-different vector the target for the relative position output was always 0, and when the input to the question layer was the relative position vector the target for the same-different output was always 0. We trained 10 instances of the relation network on two original datasets, one with the input to the question layer corresponding to the same-different vector, and one with question input corresponding to the relative position vector. For all other datasets we set the question inputs to the relative position vector.

The training setting for simulation 3b was identical to simulation 3a for both the ResNet models and the relation network, with the exception that the second output layer had 10 units (correspondin with the 10 conditions) and the second term in Equation (1) corresponded with the categorical cross-entropy instead of binary cross-entropy.

Results and discussion

Figures 18 and 19 in Appendix A present all validation AUC scores for all trained datasets per task, model and condition for simulations 3a and 3b, respectively. Figure 8 presents the results of simulation 3a. As can be seen, all models achieved ceiling performance in the relative position task. Overall, all models achieved a similar level of performance in the same-different task, with the ResNet-50 being the best performer by a small margin. The ResNet-50 and ResNet-152 performed poorly on the lines and arrows conditions, whereas the relation network performed poorly only in the arrows condition. Strikingly, the relation network performed worse than both the ResNet models in the regular condition. Figure 9

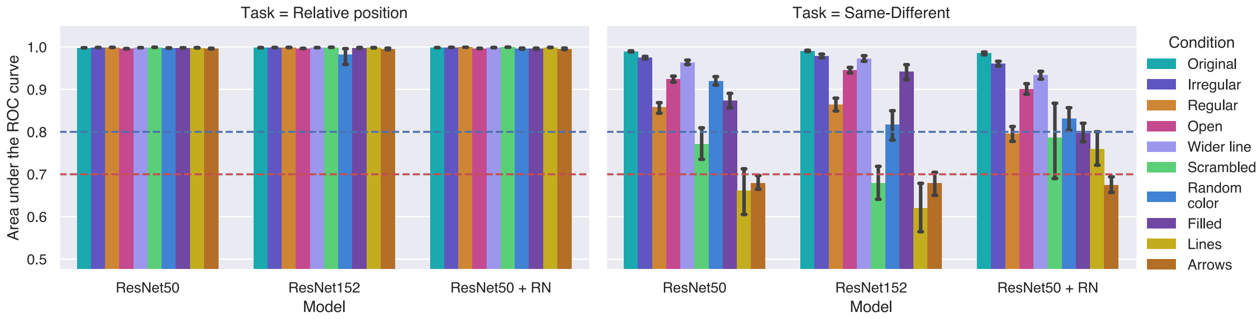


Figure 8. Mean AUC by test dataset, model, and task on simulation 3a. Error bars are 95% confidence intervals.

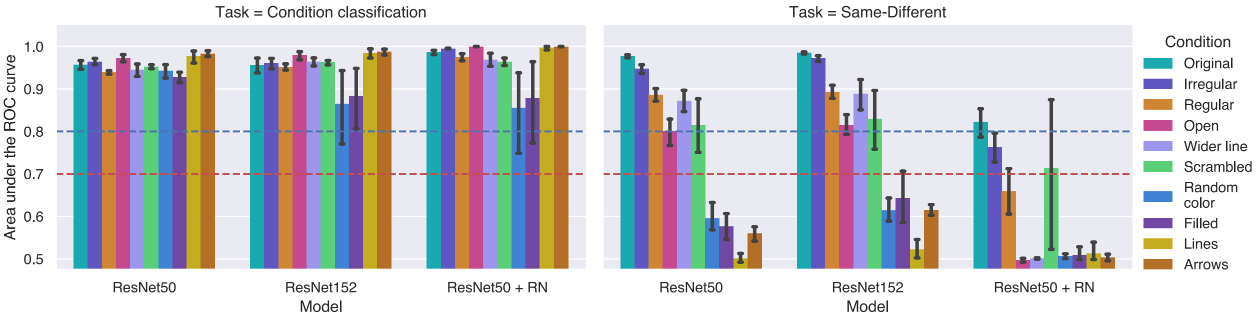


Figure 9. Mean AUC by test dataset, model and task on simulation 3b. Error bars are 95% confidence intervals.

presents the results of simulation 3b. As can be seen, the performance was worse than in simulation 3a for all models. Performance on the condition classification task was worse than in the relative position task (although all models achieved excellent performance or higher in all conditions) and performance was worse in the same-different task in comparison with simulation 3a. This drop in performance was especially marked in the relation network, which achieved excellent performance only in the trained original condition. These results show that the relative position task was better at promoting same-different generalization than the condition classification task. More generally, results of simulations 3a and 3b show that augmenting the model's experience by training on a auxiliary task can enhance generalization on the same-different task for unfamiliar samples. However, neither of our auxiliary tasks improved generalization evenly across datasets and models, which suggest that the auxiliary task is not helping the models to form an abstract representation of the relations *same* and *different*.

Simulation 4: Leave-one-out and multitask learning

In simulation 4, we combined the approaches taken in simulations 2 and 3 to provide the models with the maximum amount of information to generalize the same-different task to the unseen conditions. As in simulation 3, we trained the models in both the same-different and the relative position tasks. Furthermore, as in simulation 2, for the same-different task we trained on all the stimulus conditions except one. For each of these 9 conditions, we trained 10 instances of the ResNet-50 and ResNet-152 classifiers as well as the relation network and tested them on the stimulus set that was not trained on. We trained the our models with loss (1), this time setting w^{sd} to 1 for all datasets except the one tested on. All other training parameters were the same as in simulation 3 for both models.

Results and discussion

Figures 20 and 21 in Appendix A presents all validation AUC scores for all trained datasets per task, model and condition for simulation 4. As can be seen in Figure 10, for the relative position task all models achieved ceiling performance in all test datasets, just as in simulation 3a. For the same-different task overall performance was comparable with simulation 3a, although the performance at the condition level was different. Despite being trained on all datasets on the relative position task and on all *other* datasets (i.e., except on the dataset tested) on the same-different task, all models achieved only an acceptable level of performance on the arrows and scrambled datasets and performed poorly on the lines dataset. Overall, training on the secondary relative position task and training on the same-different task in all conditions but the one tested did improve same-different generalization,

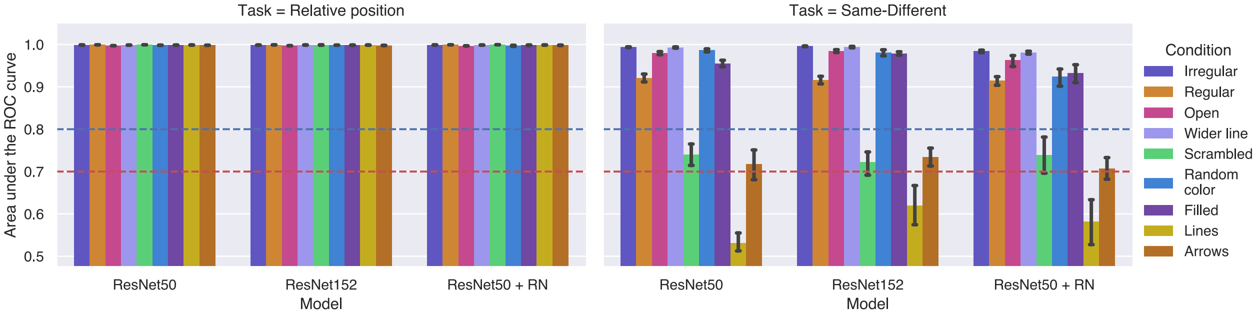


Figure 10. Mean AUC by test dataset, model and task on simulation 4. Error bars are 95% confidence intervals.

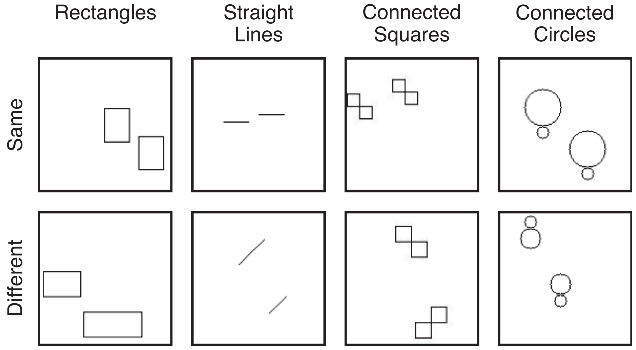


Figure 11. Examples of the same and different categories from the new testing conditions of simulation 5.

however, as in simulations 2 and 3, this effect did not spread evenly across datasets, which is not consistent with the models learning the abstract relational concepts *same* and *different*.

Simulation 5: Further tests of generalization, does a rich training regime guaranties same-different generalization?

Simulations 1 to 4 showed that augmenting the model's training regime, either directly by training on the same-different task with data from several datasets or indirectly by multitask learning, improved generalization in both the ResNet classifiers and the relation network. As described previously, however, the benefits of both strategies did not spread to all test datasets equally, which raises the question of how broadly the benefits extend. In other words, could a rich training regime that included all our above conditions lead the models to generalize same-different discrimination in yet unseen new variations of this task? To investigate this question, in simulation 5 we trained the ResNet classifiers and the relation network on all previous datasets on both tasks and tested them in four new test datasets (see Figure 11). These datasets followed the same rule as SVRT problem #1, but differed from the previous conditions at the pixel-level. They were defined as follows:

- In the rectangles dataset, each shape was a rectangle. In the same category, both shapes were identical, whereas in the different category, either the width or the height of one of the rectangles was different. Both sides had lengths between 16 and 64 pixels and the minimum difference between the critical sides on the different category, was 4 pixels.
- In the straight lines dataset, each shape was a straight line with a tilt of 0°, 45°, 90° or 135°. In the same category, both lines were identical. In the different category, one line was longer than the other. The lines had a length between 16 and 64 pixels and minimum difference in length was 4 pixels. This differences were uniformly distributed across examples of the different category.
- In the connected squares dataset, each shape was a pair of connected squares. These shapes were generated by adding an horizontal line to the shapes in the lines condition (compare the third column of Figure 11 with the eighth column of Figure 2). In the same category, both shapes were identical. In the different category, the corner at which both squares were connected was the opposite.
- In the connected circles dataset, each shape was a pair of connected circles where one was in top of the other. One the circles was bigger than the other. In the same category, both shapes were identical. In the different category, the circles at the top and bottom were swapped.

To train the models, we used the same settings as the previous simulations, except that we trained the 10 model instances for 20 epochs.

Results and discussion

Figure 22 in Appendix A presents all validation AUC scores for all trained datasets per model and stimulus condition. As can be seen in Figure 12, all models achieved ceiling performance on the relative position task in all conditions, including the new test datasets. In contrast, both models only achieved acceptable performance on the rectangles dataset and poor performance on the straight lines, connected squares and connected circles datasets in the same-different task. This marked difference in task generalization is consistent with previous results by Kim et al. (2018); see also Vaishnav et al. (2021), who reported that visual reasoning tasks that involve same-different judgments are more difficult for CNNs than other spatial reasoning tasks. Furthermore, these results provide further support to the idea that DCNNs do not form abstract representations of the relations *same* and *different* when trained on the same-different task.

Although analyses based on AUC scores provide an overall measure of the degree of generalization achieved by the models, this measure does not provide much detail about the nature of the errors—for example, whether a model is assigning high probabilities of belonging to the same category to samples of the different category (i.e., false positives), assigning low probabilities to samples of the same category (i.e., false negatives), or both. To gain further understanding of all the models' behavior in this simulation, for each model we picked the best performing instance (according to the AUC scores) and used it to plot the distribution of predicted probabilities for each test dataset (Figure 13).[4] For each model/dataset combination we also calculated the optimal threshold that maximized the true positive

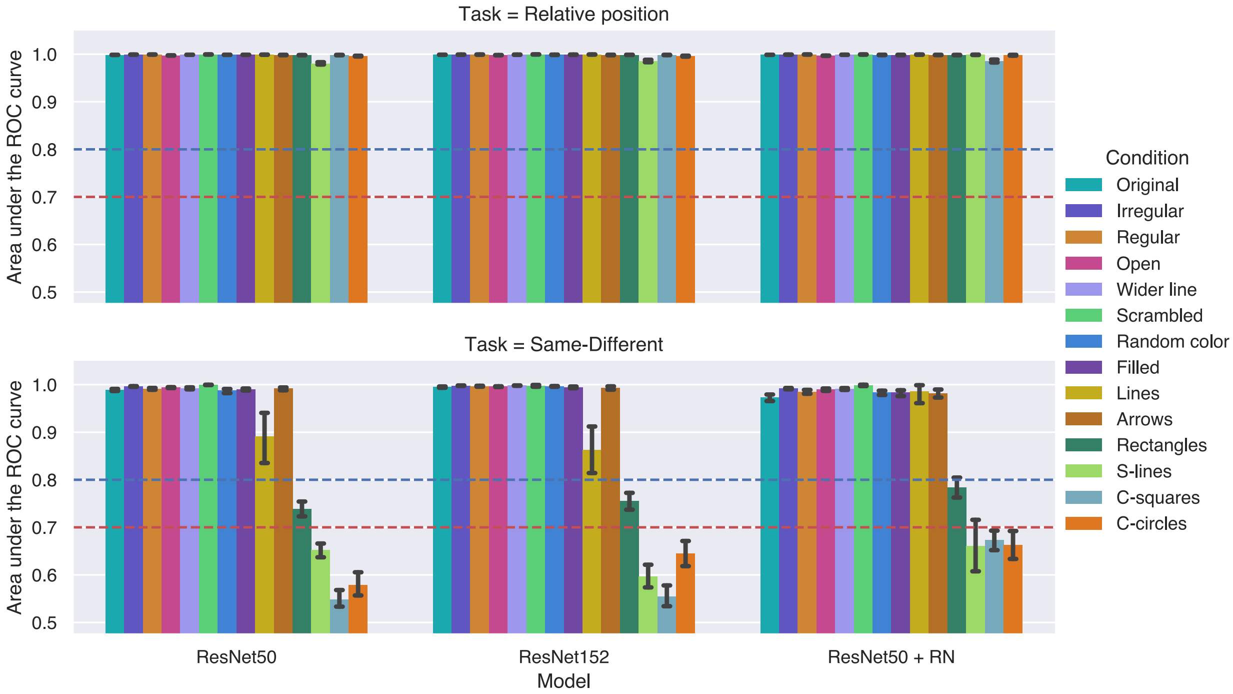


Figure 12. Mean AUC by test dataset, model and task on simulation 5. Error bars are 95% confidence intervals.

rate and the true negative rate, through Youden's J index (Youden, 1950), and superimposed on the distributions. As can be seen in Figure 13, on the trained stimulus conditions the distributions of predicted probabilities for the same and different categories showed a low degree of overlap, which allows the optimal threshold to discriminate effectively between the categories. In contrast, in the new test conditions there is a greater degree of overlap between the distributions which renders the optimal threshold ineffective. This overlap is due to a tendency of to assign high probabilities of belonging to the same category to samples of the different category; in other words, all models tend to produce false positives in the new conditions. The exception to this general pattern is the rectangles condition, where the mass of predicted probabilities was more evenly distributed across the whole range of probability values for the samples of the different category. Overall, all models showed a similar degree of overlap between the same and different categories in the new stimulus conditions, with the relation network showing a slightly lower overlap.

One comparison that is specially informative is the one between the lines and the connected squares conditions. Recall that the connected squares conditions was built by simply adding a horizontal line to the shapes in the lines condition. This simple change—that left the underlying same-different classification rule intact—lead all models to dramatically increase the degree of overlap between the probabilities assigned to the samples of the same and different categories. Overall, the results of simulation 5 show that, for all models, generalization of the same-different task was highly restricted in the case of the new untrained conditions, exactly the opposite one would expect if these models had learned the abstract relations *same* and *different*.

Simulation 6: The role of object segregation

Simulations 1 to 5 showed that DCNNs do not learn the abstract *same* and *different* relations. What is needed to accomplish this? As discussed in the introduction, (Kim et al., 2018; see also Ricci et al., 2021) the Siamese Networks (Bromley et al., 1993), a model that simulates the effects of attentional selection and perceptual grouping by encoding the two objects of the same-different examples in two separate channels, was able to solve the same-different task easily.

However, there are reasons to doubt that separating the objects of the same-different examples into different channels is all that is needed for a DCNN to learn an abstract representation of the relations *same* and *different*. First, there has been no tests of the generalization capabilities of the Siamese network on the kind challenging stimuli we have used on simulations 1 to 5. Second, recently Webb et al. (2021) have shown, using a custom dataset, that a recurrent version of the Siamese network failed on the same-different task on examples not seen during training. Third, object segregation is only one of several aspects involved in forming a relational representation like *same* or *different*. In particular, as Greff et al. (2020) have pointed out, achieving relational responding in neural networks might entail binding (already segregated)

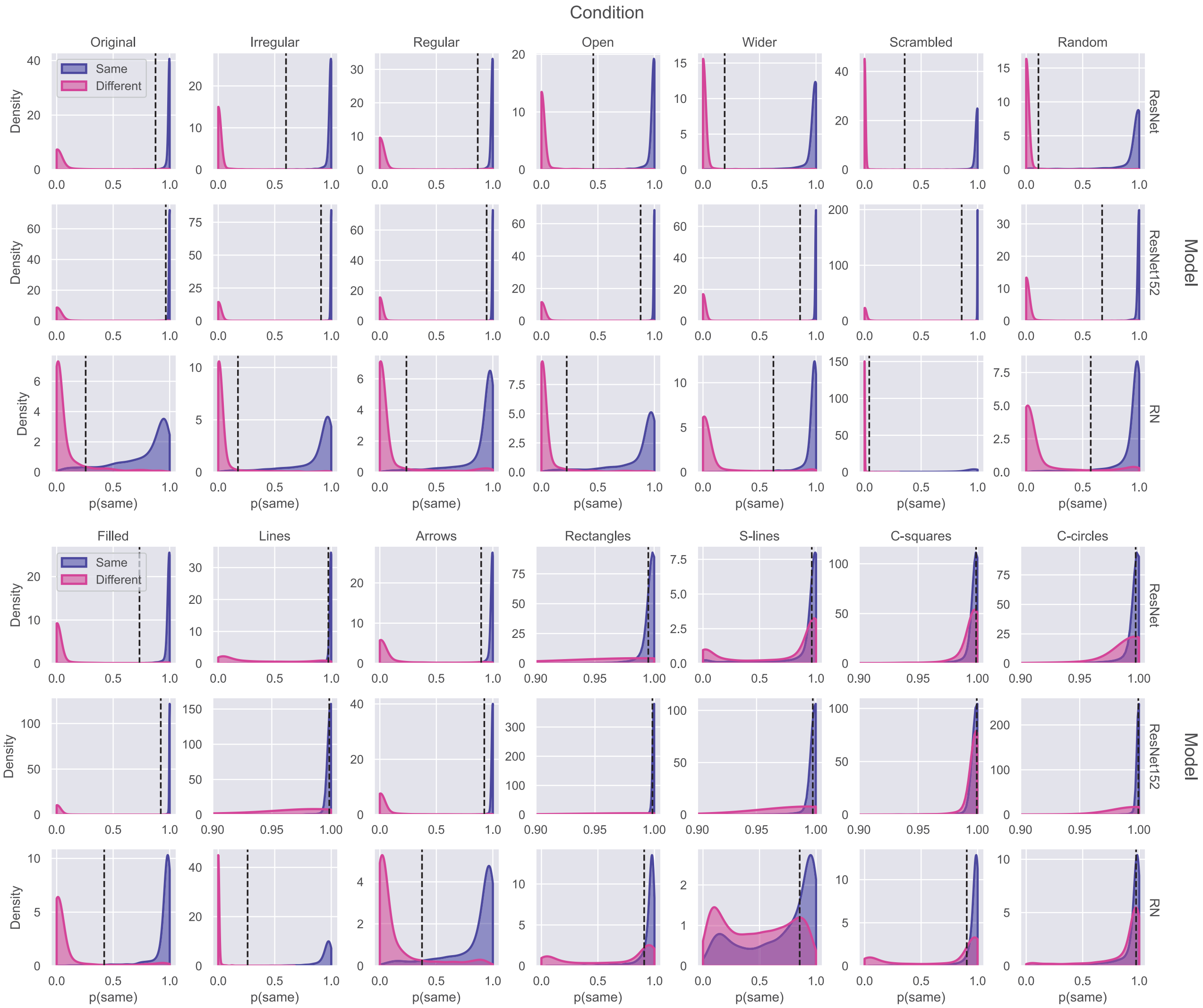


Figure 13. Gaussian kernel density estimates for test datasets on simulation 5. Dashed lines represent optimal thresholds for each model/dataset combination. See text for details.

object representations with independent representations of relational roles dynamically, something that is beyond the capabilities of standard neural networks architectures. In other words, object segregation might be only a necessary but not sufficient condition for a neural network to form relational representations.

To investigate the role of object segregation in same-different generalization, in simulation 6 we replicated simulations 1 and 5 with Siamese networks. Our models (Figure 14), use the same front-end (ResNet-50 or ResNet-152) in both channels (i.e., they share weights) to produce two vectors through a GAP operation, that are concatenated and passed to two hidden layers of ReLU units which lead to a single same-different output unit. To train and test the model, we made versions of all our new datasets with the objects separated into two images. In simulation 6a, we trained the model on the irregular dataset and tested it in all other datasets (as in simulation 1). In simulation 6b, we trained on the same-different task in all the datasets of simulation 6a and tested on the rectangles, straight lines, connected squares, and connected circles conditions (as in simulation 5; note, however, that

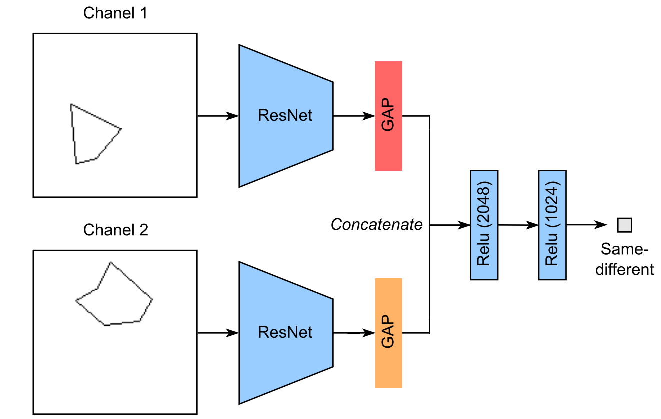


Figure 14. Siamese network.

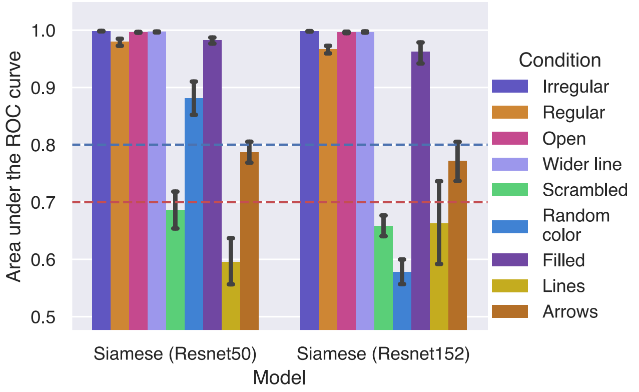


Figure 15. Mean AUC by model and condition on simulation 6a. Error bars are 95% confidence intervals.

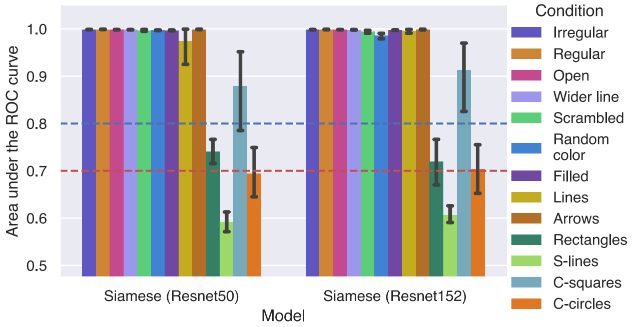


Figure 16. Mean AUC by model and condition on simulation 6b. Error bars are 95% confidence intervals.

we did not use a secondary task). We used the same training settings as the previous simulations, except that in simulation 6a we froze the front-end and trained the classifiers for three epochs and then trained the whole model for five epochs and in simulation 6b we trained the classifiers for four epochs and then the whole model for six epochs.

Results and discussion

In simulation 6a, the validation AUC was above 0.9 for both models on the irregular dataset. As can be seen in the first panel of Figure 15, simulation 6a shows that training on the separated channels version of the Irregular dataset did not produce perfect generalization to the untrained datasets for either of the models. Both models achieved similar results, with the ResNet-50 based achieving acceptable performance on the arrows dataset and poor performance on the scrambled and lines datasets. The ResNet-152 based model showed the same pattern of results except on the random color condition, where it performed poorly. Figure 23 in Appendix A presents all validation AUC scores for all trained datasets per model and stimulus condition in simulation 6b. As can be seen in Figure 16, the results of simulation 6b show that training in all previous datasets did not produce perfect same-different generalization either. It is worth noting, however, that simulation 6b produced better results than simulation 5, with performance on the connected squares dataset going from poor to excellent.

Overall, these results show that using separated channels for the two objects of each example (and therefore, assuming object segregation as given) did not produce perfect generalization on the same-different task. These results imply that object segregation is not a sufficient condition for learning the relations *same* and *different* on DCNNs. In this regard, one possibility is that, beside object individuation, it might be necessary to separate the content of the representations of the objects participating in the relation from the content of the relational roles that form the abstract relations *same* and *different*, as postulated by part-based theories of object recognition (Biederman, 1987; Hummel & Biederman, 1992) and current treatments of compositional representations on deep neural networks (Greff et al., 2020). Of course, it is still possible that some DCNN-based model could achieve abstract reasoning abilities without explicit object segregation, however, the results of simulations 1 to 5 suggest otherwise.

General discussion

In six simulations, we tested whether DCCNs were able to learn the abstract *same* and *different* relations when trained on the same-different task. Across simulations we found that, instead of forming an abstract representation of this task that generalizes beyond the training distribution, DCCNs were unable to reliably generalize to new test images that shared the same underlying relations as the training data but were dissimilar at the pixel level. This was the case even when we augmented DCCNs' experience with new stimulus sets that instantiated the same-different task with several kinds of objects (simulations 2, 4, and 5), and when we used multitask learning to give them experience with images from same distribution in a different task, and thus ensuring that the models were able to process the test stimuli (simulations 3, 4, and 5). Furthermore, in simulation 6 we showed that separating the two objects of the same-different images into different channels was not enough to enable DCNNs to learn the abstract notions of *same* and *different*, even when trained in a rich regime with data from several datasets.

These results shed new light into the discussion of whether is necessary to invoke extra, symbolic mechanism to solve the same-different task. If by "solving" the same-different task one means generalizing from one set of images to another set of images that share the same pixel-level distribution, it is perfectly reasonable to say that DCNNs are able to solve this task. This problem, by itself, is interesting from a machine learning point of view, because simpler machine learning models tested previously could not solve this kind of task. However, if by "solving" the same-different task one means to learn a representation of the *same* and *different* relations that support generalization beyond pixel-level similarity (as in humans and chimpanzees), our results suggest that DCCNs are just not up to the task and that some sort of symbolic machinery may be necessary.

Consistent with this conclusion, the relation network used in experiments 1 to 5 did not fair much better than standard CNNs classifiers when the training and test images were markedly different at the pixel level. Importantly, the relation network is claimed to support relational reasoning without implementing symbolic computations. Our results show, however, that exhaustively comparing feature columns from all locations in the output filters of a DCNN does not support out-of-distribution generalization of same-different judgements, as one would expect from a model that learned a relational representation of the same-different task. Beside failing to learn what could be considered the simplest possible visual relations, the number of parameters of the relation network grows combinatorially with the filter size of the DCNN output, which makes this model much more inefficient during training than standard CNNs classifier and brings into question the scalability of this approach.[5]

Similarly, the Siamese network failed to support same/different judgments when training and test images were from different pixel-level distributions. Clearly, object individuation is a necessary step in the process of comparing objects, but our findings highlight that hard-wiring this information in a DCNN is not sufficient in to solve the same-different task. As many have suggested (e.g., Webb et al., 2021; Hummel & Biederman, 1992; for a review, see Greff et al., 2020), for a neural network to achieve effective relational generalization, mechanisms to represent objects and relational roles independently and binding them together dynamically might be necessary. Recently, Webb et al. (2021) proposed a emergent symbol binding network (ESBN) that aims to implements some of this principles. Using a custom dataset, they show that the ESBN model was able to generalize the same-different task to completely new objects. Note, however, that 1) the ESBN model takes as inputs images of individual objects and 2) the dataset of Webb et al. (2021) was composed by combining 100 32×32 grayscale images of simple Unicode characters, which raises the question of whether the ESBN model would show the same degree of generalization with more complex stimuli like ours. We think that this is an interesting possible extension of the current research. More generally, as Stabinger et al. (2021) note, models that work with separated channels for different objects assume object segregation as given, which is one of the most important steps of the processing of visual relations. A satisfactory solution to the same-different task in neural networks should be able to extract the critical objects to compare from the image automatically.

In conclusion, our results show learning same-different relations is beyond the current capabilities of DCNNs. Fundamental work on mechanisms for object individuation and dynamic binding seems necessary for neural networks achieve this hallmark of intelligence.

Keywords: same-different relations, relational reasoning, visual relations, deep neural network

Acknowledgments

Funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 741134).

Commercial relationships: none.

Corresponding author: Guillermo Puebla.
Email: guillermo.puebla@bristol.ac.uk.
Address: School of Psychological Science, University of Bristol, 12a Priory Road, Bristol BS8 1TU, UK.

Footnotes

[1] We also made models that had the pretrained convolutional front end frozen and only the classifier was trainable. Those models achieved similar results to the ones presented on simulation 1. However, they were not well-suited for the data augmentation and multi-task learning techniques used on Simulations 2 to 4, so we do not consider them further.

[2] To validate our implementation of the relation network we replicated on Appendix B the main results of Santoro et al. (2017) with the Sort-of-CLEVR dataset.

[3] The question layer was part of the original implementation of the relation network by Santoro et al. (2017), but was unnecessary in the previous simulations because the question was constant (same or different).

[4] We used the kernel density estimate function of Seaborn (Waskom, 2021). We customized the x-axis of some of the subplots for better readability.

[5] In our simulations this made training the relational network several times slower than training a the ResNet-50 classifier using a GPU.

References

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological Review, 94*(2), 115.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a "siamese" time delay neural network. *Advances in Neural Information Processing Systems, 6*, 737–744.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255).

Eitz, M., Hays, J., & Alexa, M. (2012). How do humans sketch objects? *ACM Transactions on Graphics (Proc. SIGGRAPH), 31*(4), 44:1–44:10.

Ferry, A. L., Hespos, S. J., & Gentner, D. (2015). Prelinguistic relational concepts: Investigating analogical processing in infants. *Child Development, 86*(5), 1386–1405.

Fleuret, F., Li, T., Dubout, C., Wampler, E. K., Yantis, S., & Geman, D. (2011). Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences of the United States of America 108*(43), 17621–17625.

Funke, C. M., Borowski, J., Stosio, K., Brendel, W., Wallis, T. S., & Bethge, M. (2021). Five points to check when comparing visual perception in humans and machines. *Journal of Vision, 21*(3), 16–16.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., & Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. *Advances in Neural Information Processing Systems, 31*, 7538–7550.

Gentner, D., Shao, R., Simms, N., & Hespos, S. (2021). Learning same and different relations: Cross-species comparisons. *Current Opinion in Behavioral Sciences, 37*, 84–89.

Greff, K., van Steenkiste, S., & Schmidhuber, J. (2020). On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29–36.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778), doi:10.1109/CVPR.2016.90.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). New York: John Wiley & Sons.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2261–2269), doi:10.1109/CVPR.2017.243.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review, 99*(3), 480.

Kim, J., Ricci, M., & Serre, T. (2018). Not-so-clevr: learning same–different relations strains feedforward neural networks. *Interface Focus, 8*(4), 20180011.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.

Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L. K., & DiCarlo, J. J. (2018). Cornet: Modeling the neural mechanisms of core object recognition. *bioRxiv*, doi:10.1101/408385.

Liu, S., & Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)* (pp. 730–734), doi:10.1109/ACPR.2015.7486599.

Mehrer, J., Spoerer, C. J., Kriegeskorte, N., & Kietzmann, T. C. (2020). Individual differences among deep neural network models. *Nature Communications, 11*(1), 1–12.

Messina, N., Amato, G., Carrara, F., Gennaro, C., & Falchi, F. (2021). Solving the same-different task with convolutional neural networks. *Pattern Recognition Letters, 143*, 75–80.

O'Reilly, R., & Busby, R. (2001). Generalizable relational binding from coarse-coded distributed representations. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems* (Vol. 14, pp. 75–82). Cambridge, MA: MIT Press.

Penn, D. C., Holyoak, K. J., & Povinelli, D. J. (2008). Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences, 31*(2), 109–178.

Ricci, M., Cadène, R., & Serre, T. (2021). Same-different conceptualization: A machine vision perspective. *Current Opinion in Behavioral Sciences, 37*, 47–55.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach*. Cambridge, MA: MIT Press.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., ... Lillicrap, T. (2017). A simple neural network module for relational reasoning. *Advances in Neural Information Processing Systems, 30*, 4967–4976.

Stabinger, S., Peer, D., Piater, J., & Rodríguez-Sánchez, A. (2021). Evaluating the progress of deep learning for visual relational concepts. *Journal of Vision, 21*(11), 8–8.

Stabinger, S., Rodríguez-Sánchez, A., & Piater, J. (2016). 25 years of CNNs: Can we compare to human abstraction capabilities? In *International Conference* *on Artificial Neural Networks* (pp. 380–387). Cham: Springer.

St. John, M. F. (1992). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science, 16*(2), 271–306.

Vaishnav, M., Cadene, R., Alamia, A., Linsley, D., Vanrullen, R., & Serre, T. (2021). Understanding the computational demands underlying visual reasoning. arXiv preprint arXiv:2108.03603.

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021.

Webb, T. W., Sinha, I., & Cohen, J. (2021). Emer gent symbols through binding in external memory. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=LSFCEb3GYU7.

Yan, Z., & Zhou, X. S. (2017). How intelligent are convolutional neural networks? arXiv preprint arXiv:1709.06126.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer, 3*(1), 32–35.

Appendix A

In this appendix we provide detailed validation AUC scores for all simulation where we trained the models in more than one dataset. Note that in simulations 2, 3 and 4 there are missing bars because we did not train the models in those particular datasets. In all figures we superimpose the “acceptable” and “excellent” limits defined in Table 1.

Appendix B

We benchmarked the relation network used on simulation 1 on the Sort-of-CLEVR dataset (Santoro et al., 2017, Figure 24). This dataset consists of images with six objects. Each object has a unique color (red, green, blue, orange, gray, or yellow) and it has a square or a circular shape. Each image has 20 associated questions, 10 of which are nonrelational and 10 are relational. The nonrelational questions ask for a) the shape of an object, b) the horizontal location of an object (left or right), or c) the vertical location of an object (upside or downside). These questions are considered nonrelational because to answer them a model needs to focus only on a single object. In contrast, the relational questions require the models to consider relations between the objects in the image. These questions ask for a) the shape of the object which is closest to certain object, b) the shape of the object which is furthest away from certain object, and c) the number of objects that have the same shape as certain object. The dataset consisted of 10,000 randomly generated (image, questions, answers) triplets, of which 200 were withhold for testing. To benchmark the ResNet-50 based relation network, we made images of size 128 × 128 instead of 75 × 75 as in the original dataset.

We benchmarked three models. The first two were the same models tested in Santoro et al. (2017): a four-later CNN front end with a MLP classifier (CNN+MLP) and the original relation network. The third one was the relation network with 8 × 8 inputs from `Resnet-50`. All models were trained with the Adam optimizer with a learning rate of 0.00025 for the first two models and 0.0001 for the last. As can be seen in Figure 25, both versions of the relation network achieved high levels of performance in the nonrelational and relational questions. The CNN+MLP model performed comparatively worse in both types of questions. In contrast to the results of Santoro et al. (2017), the CNN+MLP model performed better in the relational questions than in the non-relational ones. It is worth noting that the sort-of-CLEVR dataset does not include a set of withhold objects, which prevents to perform the kind of relational generalization tests that we carry on the main article. In fact, when Kim et al. (2018) tested the relation network on a same-different dataset with withhold color/shape combinations the model performed at chance.

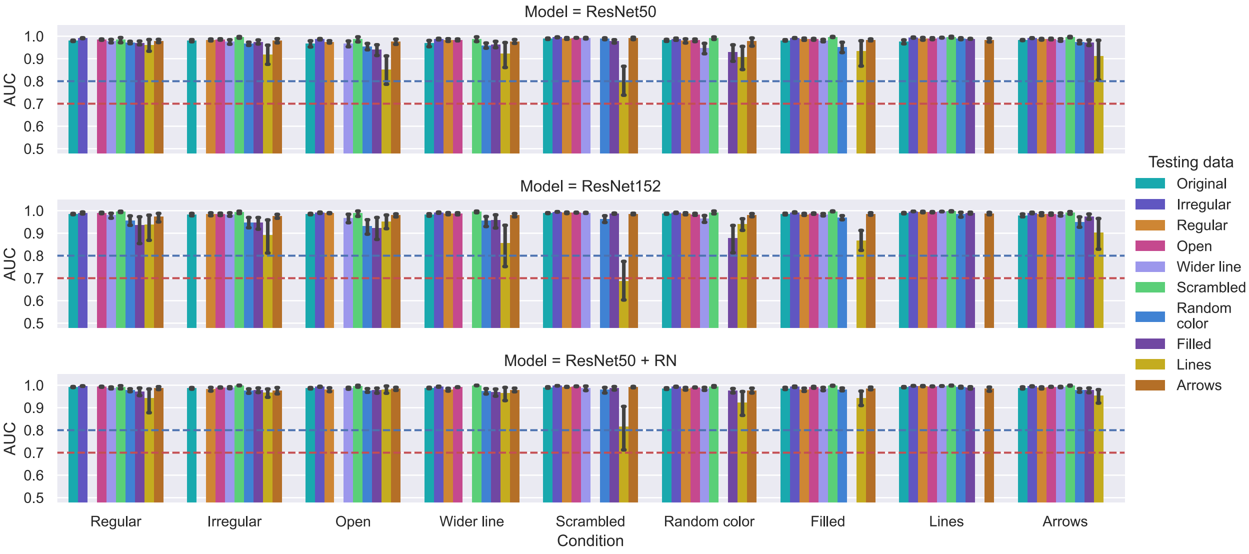


Figure 17. Mean validation AUC by model, condition and test dataset on simulation 2. Error bars are 95% confidence intervals.

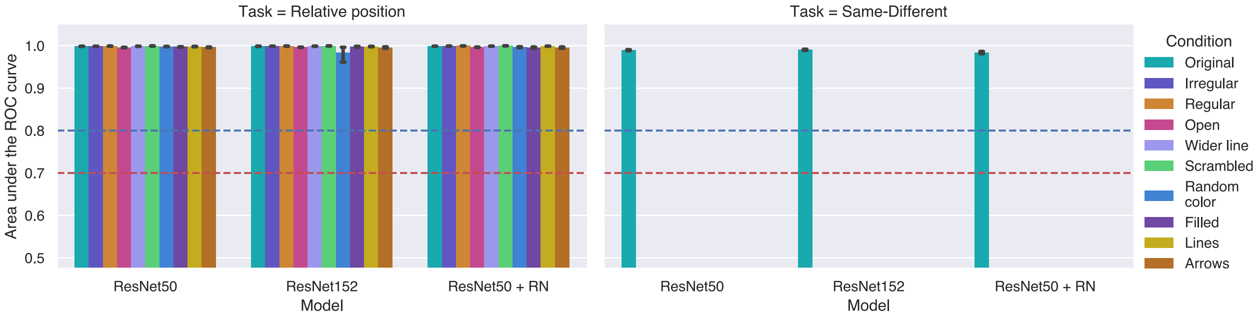


Figure 18. Mean validation AUC by task, model and condition on simulation 3a. Error bars are 95% confidence intervals.

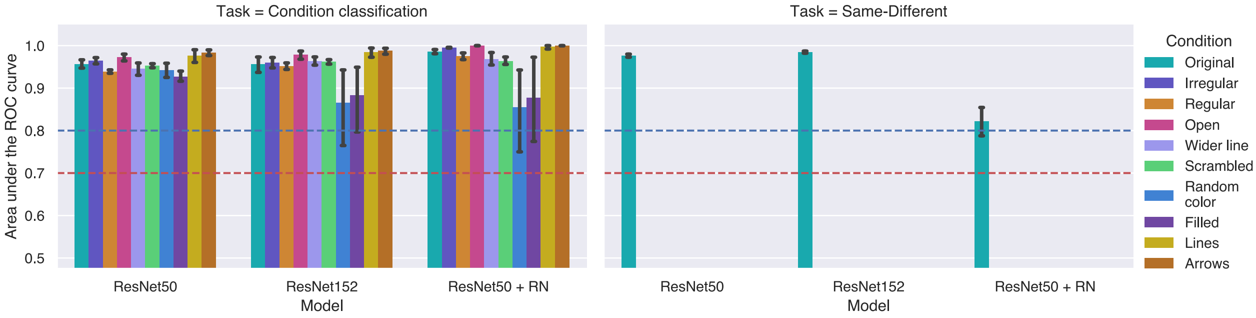


Figure 19. Mean validation AUC by task, model and condition on simulation 3b. Error bars are 95% confidence intervals.

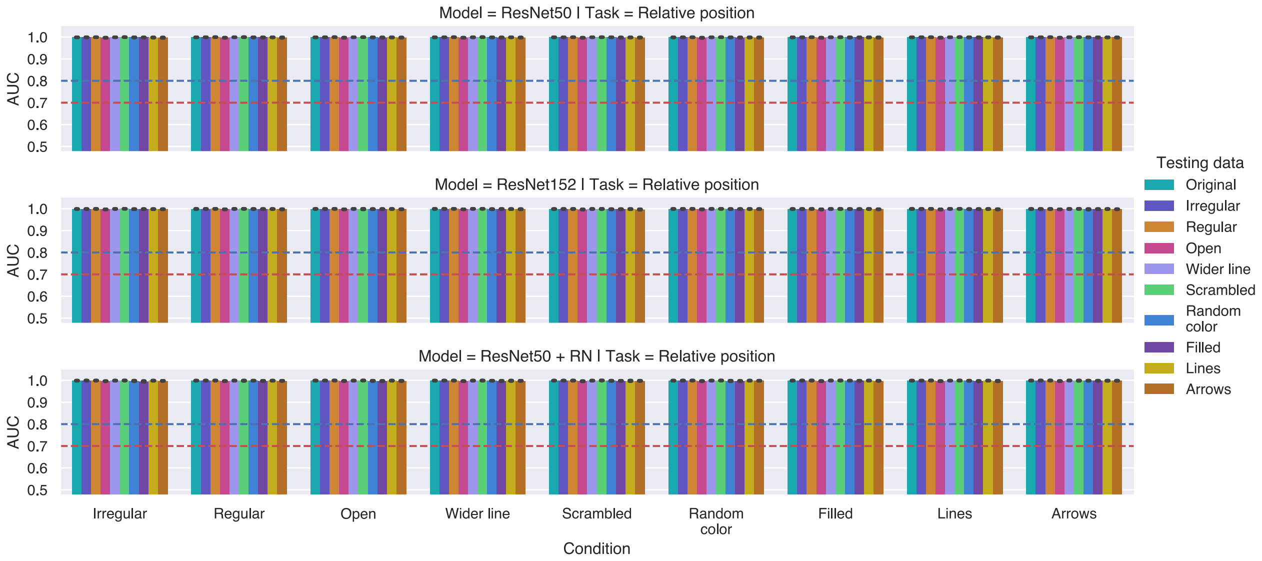


Figure 20. Mean validation AUC on the relative position task by model and condition on simulation 4. Error bars are 95% confidence intervals.

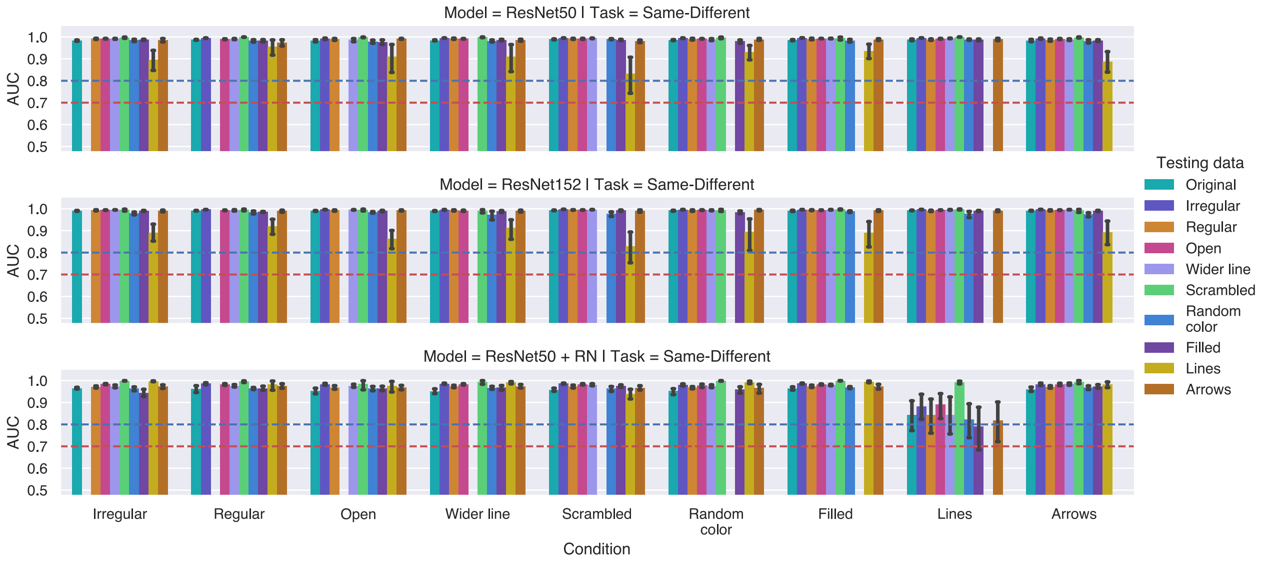


Figure 21. Mean validation AUC on the same-different task by model and condition on simulation 4. Error bars are 95% confidence intervals.

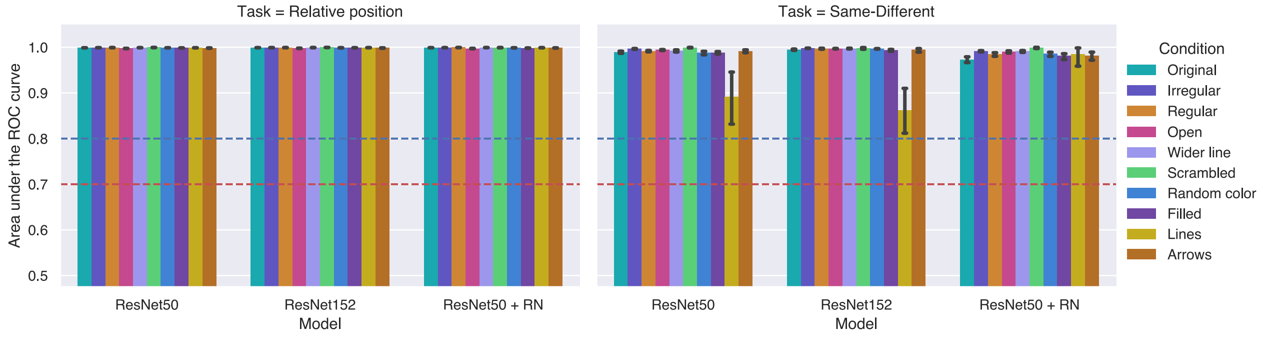


Figure 22. Mean validation AUC by task, model and stimulus condition on simulation 5. Error bars are 95% confidence intervals.

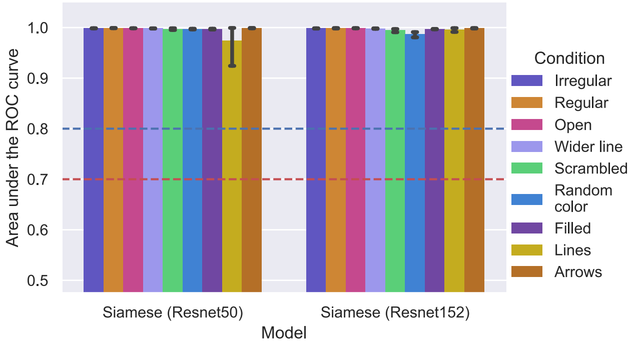
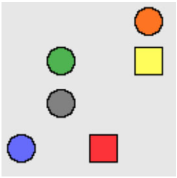


Figure 23. Mean validation AUC by model and Condition on simulation 6b. Error bars are 95% confidence intervals.



Nonrelational question

```
Q: What is the shape of the red object?
A: Square
```

Relational question

```
Q: How many objects have the same shape
as the green object?
A: 4
```

Figure 24. Example image, questions and anwsers from the Sort-of-CLEVR dataset.

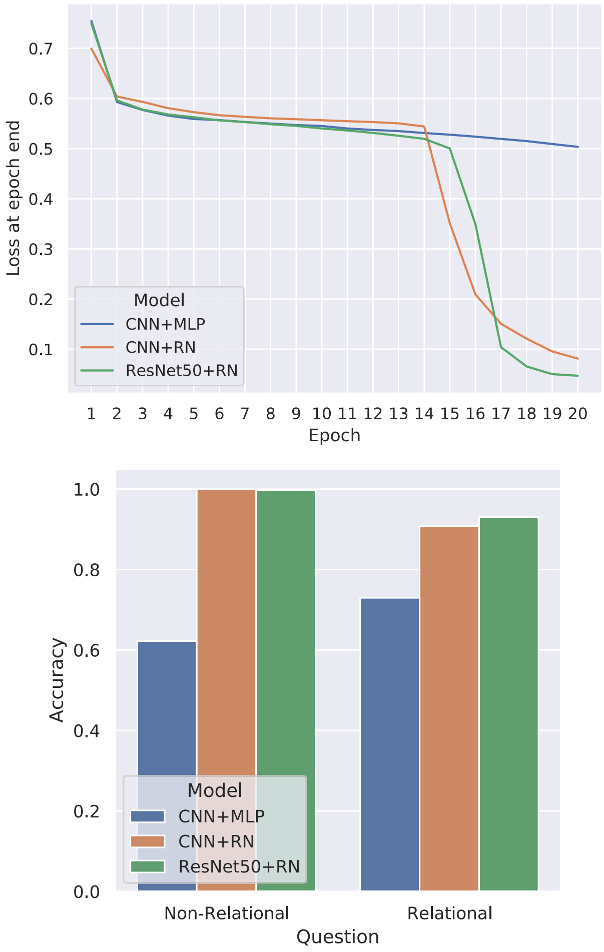


Figure 25. Sort-of-CLEVR benchmark. (Left) Training loss. (Right) Test accuracy on nonrelational and relational questions.