Evaluating the progress of deep learning for visual relational concepts

Sebastian Stabinger Universität Innsbruck, Innsbruck, Austria

David Peer Universität Innsbruck, Innsbruck, Austria

Justus Piater Universität Innsbruck, Innsbruck, Austria

Antonio Rodríguez-Sánchez Universität Innsbruck, Innsbruck, Austria

Convolutional neural networks have become the state-of-the-art method for image classification in the last 10 years. Despite the fact that they achieve superhuman classification accuracy on many popular datasets, they often perform much worse on more abstract image classification tasks. We will show that these difficult tasks are linked to relational concepts from cognitive psychology and that despite progress over the last few years, such relational reasoning tasks still remain difficult for current neural network architectures. We will review deep learning research that is linked to relational concept learning, even if it was not originally presented from this angle. Reviewing the current literature, we will argue that some form of attention will be an important component of future systems to solve relational tasks. In addition, we will point out the shortcomings of currently used datasets, and we will recommend steps to make future datasets more relevant for testing systems on relational reasoning.

Introduction

Convolutional neural networks (CNNs) have become the go-to method for image classification since Krizhevsky, Sutskever, and Hinton (2012) were able to win the ImageNet competition (Deng et al., 2009) by a wide margin. Despite the success of CNNs in the field of image classification, there remain some classification problems that seem to be much more challenging for CNNs and other currently available neural network architectures. Examples for such challenging tasks can be found in a subset of the Synthetic Visual Reasoning Test (SVRT) dataset by Fleuret et al. (2011) or in work inspired by Raven's Progressive Matrices (Raven et al., 1938). In this article, we will try to convince the reader that tasks that can be categorized as relational concepts are relevant for practical applications and are still difficult to solve for currently used deep learning architectures. In addition, we will point out that all currently used datasets to test for relational reasoning have one shortcoming or another. Our hypotheses, which we will try to argue for in this work, are as follows:

Hypothesis 1 (H1). Attentional mechanisms will be an important component to successfully and efficiently learn relational concepts.

Hypothesis 2 (H2). Relational concepts are more difficult to learn for current neural network architectures than other concepts.

Deep learning (LeCun, Bengio, & Hinton, 2015) has become the workhorse of the machine learning community in the last 10 years. In the form of CNNs, first introduced by LeCun et al. (1989), it has been especially successful in solving computer vision tasks. Deep learning systems are built from networks of artificial "neurons." In the end, such a neuron is a weighted sum of its inputs x with an added bias b. The weights w of this sum are also said to be the weights of the neuron. A nonlinear function f (called the activation function) is then applied to the resulting sum:

$$y(x_1, x_2, ..., x_n) = f\left(\sum_{j=1}^{n}(x_j w_j + b)\right) \quad (1)$$

These neurons are generally organized in layers and the outputs of neurons of previous layers become the inputs of neurons of later layers. The weights, therefore, connect neurons of the previous layer to the neurons of the next layer. Deep learning is named after the circumstance that modern architectures

Citation: 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, 1–23, https://doi.org/10.1167/jov.21.11.8.

https://doi.org/10.1167/jov.21.11.8 Received February 27, 2021; published October 12, 2021 ISSN 1534-7362 Copyright 2021 The Authors

are, in comparison to previously used artificial neural networks, very deep (i.e., they are constructed of many, sometimes up to multiple hundred, layers)

The output of a neuron, for a given input, can be changed by modifying its weights w. The main idea of deep learning systems is to change the weights of all neurons so that a given input will produce an expected output. Changing the weights by hand is not feasible, so training data are utilized to automatically adapt the weights. Data used for training the network consist of input/output pairs (e.g., images and their correct class). The network is fed with the inputs and a loss function is used to compare the output of the network with the known correct output. More specifically, the loss function returns a numeric value that indicates how close the output of the network is to the correct output.

This whole neural network is in essence a giant formula that is differentiable. Single points of nondifferentiability generally do not pose a problem in practice and occur in many modern neural network architectures.[1] Therefore, we can also calculate the gradient of this whole system with respect to the weights. The gradient gives us information on how to change the weights so that the output of the loss function will get slightly lower for the currently presented data. Doing this over and over for known data moves the weights of the neural network to values that will give us the correct output for a certain input.

If the training was done successfully and with enough and representative training data, the resulting network will now also generalize to unseen data and will give us the correct output for previously unseen input (e.g., the correct class for a new image that is not in our training data). This, of course, only works within limits (i.e., if the training data are statistically similar enough to the unseen data so that the network can generalize). Mårtensson et al. (2020) could, for example, show that for MRI images, even different tissue contrast from the training set leads to much poorer generalization performance. The question of why deep neural networks generalize as well as they do in many cases and at which point generalization breaks down are questions that are not yet answered sufficiently and are still a topic of ongoing research (e.g., Zhang, Bengio, Hardt, Recht, & Vinyals, 2017, and Novak, Bahri, Abolafia, Pennington, & Sohl-Dickstein, 2018, among many others).

Concept learning

Concepts are the glue that holds our mental world together (Murphy, 2004).

The decision of which group (or class) a stimulus belongs to is usually called *classification* in the field of machine learning. In cognitive psychology, the same task is more widely known as *categorization* and is thought to be facilitated by knowledge in the form of concepts.

The idea of concepts emerged from the observation that humans categorize and group objects and experiences to be able to efficiently navigate the world and transfer knowledge from one concrete physical object to another. For example, although every object in a grocery store is unique, we might categorize multiple of them into the concept of “Tomatoes” and transfer the knowledge we have obtained from past experiences with other objects in the concept “Tomatoes,” and even information we have read about the concept “Tomatoes,” to infer a lot of information about other concrete physical objects (i.e., other tomatoes) that we have never seen before. Therefore, the concept “Tomatoes” allows us to infer that we probably would or would not like to eat these concrete tomatoes, although we have never tasted them. Being able to form vast networks and hierarchies of robust concepts is what allows humans to successfully navigate even completely foreign environments and situations. The ability to learn such concepts from observation and experience is called *concept learning*. From the point of view of machine learning, concept learning is therefore all about maximal utilization of and generalization across training data. Three broad types of concepts can be differentiated, namely, *perceptual*, *associative*, and *relational* concepts[2]:

Perceptual concepts, also known as similarity-based concepts, group stimuli by their physical similarity. The perceptual concept “tree,” for example, can be learned by the fact that most trees look similar (i.e., the statistical distribution of features of one tree is similar to the statistical distribution of features of another tree).

Associative concepts emerge because multiple stimuli are associated with the same event or outcome. Thus, one member of an associative class can be represented by another member of the same class. A human can, for example, associate the written word “tree,” the picture of a tree, and an actual physical tree, because all these stimuli convey the same abstract meaning (i.e., in many contexts, the word “tree,” the picture of a tree, and an actual physical tree can stand in for each other). This is, for example, what allows humans to transfer knowledge gained by reading about trees to actual physical trees.

Relational concepts put multiple entities in a relationship to each other. The same–different concept is one of the most studied relational concepts. For a human, it is very natural to attach the label “same” to objects if they are similar in some property (e.g., height, color, movement direction). It is essential to differentiate between *perceptual* and *relational* concepts: A cup might be grouped into the perceptual concept “cup” because it looks similar to other cups. Given a scene with multiple cups, a subset of these cups might be grouped if they are more similar to each other than they are to the other cups, and the relational concept "similar cups" might be applied to this group of cups. This information might, for example, be used to determine that all of those "similar cups" probably are able to hold the same amount of liquid, without having to actually test each individual cup. Another kind of relational concept includes transitive relations like "stronger than," which can be used to infer a strength hierarchy without having to test one's strength against every member of a group. If A is stronger than I, and B is stronger than A, it is very likely that B will also be stronger than myself, so I can prevent possible injury by not even competing against B. So a perceptual concept can apply to a single entity, while a relational concept can only be applied to at least two entities.

The rest of this article is structured as follows: In the next section, "Related work," we will present experimental evidence that many animals are also able to learn the previously mentioned concept classes, give an overview of often used neural network architectures, and provide an overview of how existing research in the area of deep learning relates to these concept classes. The section "Current research on deep learning for visual relational concepts" composes the main part of this article. We will take a closer look at deep learning research on relational concepts and showcase that deep learning methods still struggle with such tasks for the most part. In cases where deep learning systems seemingly are able to solve relational concept tasks, we will point out possible flaws in the datasets indicating that it is difficult to conclude whether the neural network learned the real underlying relational concept with currently used datasets. The discussion will try to coalesce all the findings into actionable steps for further research.

Related work

The idea of concepts as well as the three classes of *perceptual*, *associative*, and *relational* concepts emerged from an anthropocentric perspective. Therefore, it is not surprising that humans have no difficulty learning all of them. However, there is also sufficient evidence that at least some animals can learn these concepts to some degree. This indicates that the ability to form abstractions, separate from concrete physical objects, is not something that only humans possess and, more important, that not only humans can learn.

Regarding **perceptual concepts**, Herrnstein and Loveland (1964) did already show that pigeons can be trained to classify images (e.g., into the classes "person" and "nonperson") and also generalize to new, unseen images, indicating that they can learn perceptual concepts. Schrier and Brady (1987) showed the same for macaque monkeys, Vogels (1999) for rhesus monkeys, Vonk and MacDonald (2002) for gorillas, and Vonk and MacDonald (2004) for orangutans. This might not be too surprising, since a bird will readily eat a cherry without having first tasted this specific cherry, but it shows that these concepts do not have to be genetically predetermined but can be learned from experience, even in animals.

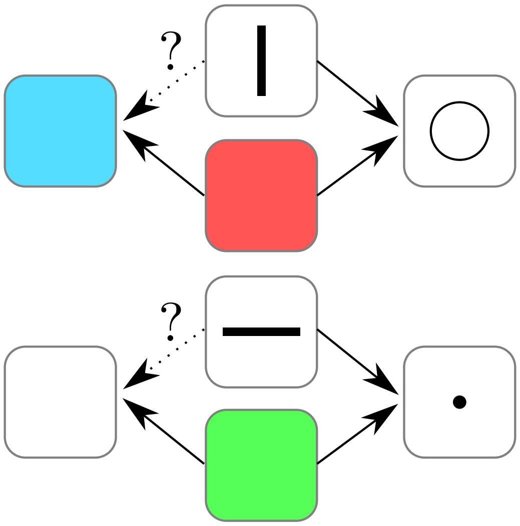


Figure 1. Visualization of a testing procedure employed by Wasserman et al. (1992) to determine whether an animal can learn associative concepts. The animal is trained to select the same response for multiple stimuli (a big circle when shown a vertical line or the color red and a small dot when shown a horizontal line or the color green). The colors red and green are later also associated with different responses (a blue light and a white light, respectively). The animal is then tested for the remaining two stimuli and the new responses. If the animal did indeed learn associative groups, one would expect that the blue light is selected for a vertical line and the white light is selected for the horizontal line, even though these specific stimuli/response pairs were never presented during training.

To test whether animals can form **associative concepts**, they can be trained to select the same response for multiple stimuli. An animal can, for example, be trained to respond to the color red as well as the picture of a vertical line by selecting a big circle. Similarly, a green light and a horizontal line can be associated with a small circle (see Figure 1). The hypothesis is that the red light and vertical line, as well as the green light and the horizontal line, would be grouped in two associative classes because they are linked to the same response. To test whether this hypothesis is correct, the red and green light are later associated with another pair of responses, namely, a blue and white light. If associative classes are formed by the animal, testing the vertical and horizontal line as a stimulus and the blue and white light as possible responses should lead to a higher probability of pairing the vertical line with the blue light and the horizontal line with the white light, even though these stimuli/response pairs were never seen by the animal before. Wasserman et al. (1992) performed exactly this experiment and could show that pigeons can learn associative concepts. This might be a more surprising outcome, but it demonstrates that even associating physically completely unrelated stimuli to each other, and therefore being able to transfer knowledge gained from one stimulus to the other, is not something uniquely human. From an evolutionary perspective, it makes sense for animals to possess the ability to form associative concepts since it reduces the amount of potentially fatal experiences an animal has to have to learn. This ability is brought to perfection in humans who can learn from the experiences of other humans by communicating abstract concepts.

The same/different task, in which stimuli have to be compared for identity or similarity in one form or another, has been the most thoroughly studied **relational concept** in animals. Zentall and Hogan (1976) showed that pigeons could choose a shape that is identical to a previously presented shape and that this ability also transfers to shapes not seen during training. These results for pigeons have been confirmed multiple times by different researchers in the following years (e.g., Blaisdell & Cook, 2005; Katz & Wright, 2006). The ability to learn the same/different concept has also been shown for bottlenose dolphins by Mercado et al. (2000); for infant chimpanzees by Oden, Thompson, and Premack (1990); for African gray parrots by Pepperberg (1987); for rhesus and capuchin monkeys by Wright and Katz (2006); for dogs by Byosiere, Feng, Chouinard, Howell, and Bennett (2017); for rats by Wasserman, Castro, and Freeman (2012); for ducklings by Martinho and Kacelnik (2016); and for bees by Giurfa, Zhang, Jenett, Menzel, and Srinivasan (2001). This allows animals to transfer information about a concrete object to similar objects and therefore make learning more efficient. In addition, researchers were able to determine that a wide array of animals are able to use transitive relational concepts to efficiently determine social order (e.g., Grosenick, Clement, & Fernald, 2007, for fish; Hogue, Beaugrand, & Laguë, 1996, for hens; and Bond, Kamil, Balda, et al., 2004, for pinyon jays).

The fact that many animals can learn concepts from all three concept classes (including relational concepts) suggests that this capability is valuable for agents interacting with and learning from the real world.

Concepts and deep learning

The question of which of the three concept classes can be learned with deep learning has not been systematically studied until now. More generally, to the best of our knowledge, the connection between concept classes and deep learning has not yet been made to the extent presented in this work. We think that this novel viewpoint is useful since the concept classes seem to align quite well with how difficult tasks are for feed-forward networks. Specifically, tasks that can be seen as learning relational concepts seem to be more difficult to feed-forward neural networks than tasks including other concepts.

Although it is rarely presented from this perspective, CNNs were specifically developed to solve **perceptual concept learning**. The architecture of CNNs is specifically designed to classify images using statistical correlations between image patterns of a more and more abstract nature as the information flows to higher layers (Cammarata et al., 2020). The tasks for which CNNs are most widely used (i.e., classifying novel images that were not seen during training) are almost identical to the experiments used to show the ability of perceptual concept learning in animals.

One widely used dataset for classification in deep learning research is the one employed in the ImageNet Large Scale Visual Recognition Challenge (Deng et al., 2009), consisting of 1.2 million training images, categorized into 1,000 classes. The top-5 error rate[3] of humans on this dataset is 5.1%, according to Russakovsky et al. (2015). It should be noted that this number was obtained by only testing a single subject, but it is the only officially published result for humans. The tested subject describes his experience with the task in Karpathy (2014). According to the author, it is difficult to even get an overview of what 1,000 classes are available for selection, and fine-grained classification (there are more than 120 different breeds of dog as separate classes in the dataset) is quite difficult for humans. The CNN architecture presented by He, Zhang, Ren, and Sun (2015) first outperformed the 5.1% error rate of the tested human subject with a top-5 error rate of 4.94%, which has steadily fallen to around 1.2% by 2020 (Pham, Dai, Xie, Luong, & Le, 2020). Considering that CNNs perform better than humans on many tasks that are similar to the ones intended to detect perceptual concept learning in animals, it is not unreasonable to assume that perceptual concept learning is the prime example of a task that CNNs are exceptionally good at.

To the best of our knowledge, deep learning has never been explicitly tested on **associative concept learning** in the way animals are usually tested. Mondragón, Alonso, and Kokkola (2017) proposed the use of deep learning architectures to model associative concept learning but did not perform any experiments.

Despite the lack of explicit experiments in this area, some research does show that a form of associative concepts emerges implicitly in certain circumstances via so-called multimodal neurons. Kim, Hannan, and Kenyon (2018a) were able to show that a biologically inspired, hierarchical CNN, which utilizes sparse coding, produces neurons that strongly activate for persons, even in costume, and the names of those persons. The authors used the biologically inspired "Locally Competitive Algorithm" by Rozell, Johnson, Baraniuk, and Olshausen (2007) to train the network. Other evidence for multimodal neurons comes from research by Goh et al. (2021) on neurons in the CLIP architecture by Radford et al. (2021), which is simultaneously trained on images and image captions using a variant of stochastic gradient descent (SGD). The authors were able to find neurons in the CLIP architecture that, for example, strongly activate for images of spiders, spider webs, and Spiderman in his costume but also images that contain the text "Spider." These multimodal neurons could therefore be interpreted as implicitly learning something akin to associative concepts.

This would indicate that at least some deep learning architectures can implicitly learn associative concepts in their intermediate representations. A direct investigation instead of purely coincidental evidence will be needed to get a better understanding of how well different deep learning architectures can deal with associative concepts.

Relational concepts are interesting since they are not the kind of problems that deep learning was initially conceived for but are nonetheless very important from a practical point of view for a wide range of computer vision applications. Imagine a robot that is asked to pass the "large cup." The visual reasoning system of this robot has to be able first to detect cups in its vicinity using perceptual concepts and then use relational concepts to compare the size of the cups to see which one might be considered the "large cup." Understanding relational concepts would also allow the robot to transfer knowledge gained about one of the cups to all similar cups, ensuring that training data are utilized most efficiently. In addition, a robot should be able to learn new relational concepts from interactions with humans. Once natural language interfaces to computer systems become commonplace, it will be essential to understand relational concepts since a sizable part of human communication utilizes relations. Therefore, it is not surprising that a lot of the research into relational concept learning (even under a different name) comes from the field of visual question answering (VQA) (Wu et al., 2017). For these tasks, a system tries to learn how to answer questions about an image, where the questions are asked in the form of natural text. These tasks, unfortunately, mix pure learning of relational concepts with problems from natural language processing (i.e., to understand the question). Therefore, we excluded VQA research from this article since it mixes two separate problem fields, which makes answering the question of whether a system could learn relational concepts even harder than it already is when concentrating on more abstract classification tasks.

Fortunately, over the last few years, researchers have looked at learning relational concepts from images using more abstract tasks to accurately measure the performance of deep learning methods on relational concept learning while minimizing the influence of other confounding factors. In this article, we will concentrate on such "pure" tasks.

Deep learning architectures

Since some specific neural network architectures were used in multiple works that will be presented in this article, we will briefly give an overview of how they work and why they might be used in certain circumstances:

CNNs: When applied to image data, the inputs of a neuron can be organized so that the output of the neuron is equivalent to the application of a filter (e.g., Gabor filters) to a specific image region. The kernel of the filter directly corresponds to the weights of the neuron. Since, in most cases, a filter for one region of an image will be equally helpful for other regions, applying the same "filter neuron" for all positions of an image is common. This procedure is equivalent to a convolution between the filter kernel and the image. Hence, layers of such neurons are called convolutional layers and neural networks making use of such layers are called CNNs. In essence, a CNN is purposefully designed to efficiently process and learn from two-dimensional data and utilize spatial invariance, which is present in many images to a certain degree.

Note that learning the weights means that the kernels of the filters used in a CNN are also learned from the data and are not predetermined. For CNNs, it has been shown that this training procedure leads to the layers extracting features, which are then combined into more and more complex features as the information flows toward higher layers. This hierarchical extraction of features has been demonstrated exceptionally well in a series of articles by Cammarata et al. (2020).

Residual networks (ResNets), introduced by He, Zhang, Ren, and Sun (2016), are one of the standard CNN architectures widely used in practice because they overcome one shortcoming of plain CNN architectures: The expressivity of a neural network (i.e., the complexity of the computed function) grows exponentially with the number of layers, but only linearly with the number of trainable parameters. This has been shown theoretically for fully connected networks by Raghu, Poole, Kleinberg, Ganguli, and Sohl-Dickstein (2017), and empirical evidence shows that this likely also holds for CNNs. So deeper networks would generally be preferred to shallower ones. Unfortunately, just stacking more layers leads to the so-called *degradation problem*, where the accuracy a network achieves when being trained on a specific

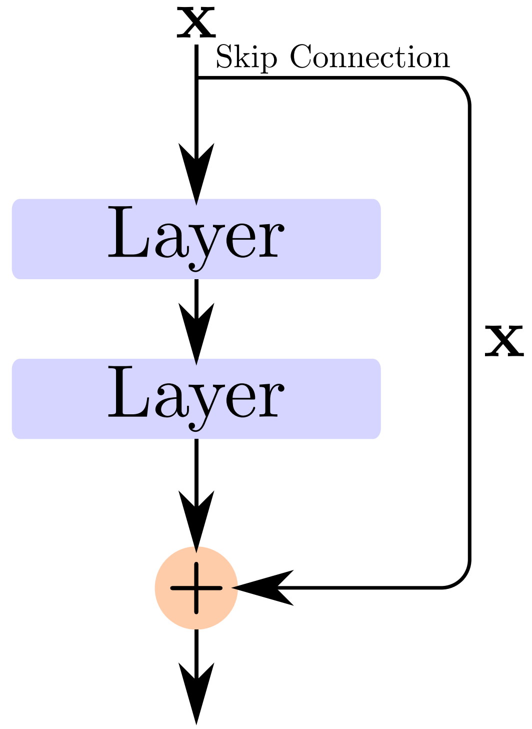


Figure 2. Schematic visualization of a residual block (the main building block of residual networks).

dataset is getting worse the deeper the network gets. This is somewhat counterintuitive since unneeded layers could just be optimized to resemble an identity function, resulting in an output identical to a shallower network. However, this does not happen in practice, indicating that deeper networks are generally harder to optimize if their architecture is not adapted.

Residual networks mitigate this problem by not only sending an input $\mathbf{x}$ through some of the network layers themselves but also adding the input to the output of the layers at a later point (see Figure 2). This forwarding of the input to deeper layers is called a shortcut or skip connection. If the layers themselves calculate $\mathcal{F}(\mathbf{x})$, this whole block calculates $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ and is called a *residual block*. By optimizing the weights of the layers, we are optimizing a residual term, hence the name residual networks.

For the standard residual networks, the layers themselves are convolutional layers, and the whole network consists mainly of a sequence of such residual blocks shown in Figure 2. Although such a residual block should theoretically not be able to learn more than the same network without the skip connection, currently used optimization schemes seem to have a much easier time optimizing this alternative residual rephrasing of the original problem. One reason is that instead of learning an identity function, the layers in a residual block only have to be pushed to output zero since the skip connection already implements the identity function. Another advantage might be that there is always one path for the training signal (via the gradient) to flow to higher layers without going through all the layers themselves.

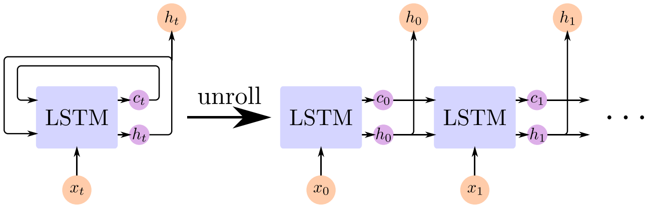


Figure 3. Schematic visualization of how an LSTM network is being applied to a sequence of inputs. On the left side is the general architecture, which is applied iteratively to the input sequence. The right part demonstrates how this iterative architecture can be unrolled to accept a whole sequence at once.

In Peer, Stabinger, and Rodríguez-Sánchez (2021), we were able to present another reason why such skip connections improve the training outcome. We were able to detect layers in neural networks that we named *conflicting layers*, where inputs with different labels collapse to a single point in the activation vector space. We showed theoretically and empirically that conflicting layers degenerate the gradient during training so that weights of the neural network are updated into wrong directions, leading to worse training outcomes. We could show that residual connections skip these conflicting layers.

All these reasons might explain why skip connections seem to perform well in practice and are among the standard architectural components of most modern deep neural networks. Because of this, residual networks have become one of the most widely used architectures for computer vision applications.

Long short-term memory networks (LSTM-networks) are a type of neural network architecture developed by Hochreiter and Schmidhuber (1997) for processing sequences of inputs and are an example of so-called recurrent neural networks (RNNs) in comparison to feed-forward neural networks like CNNs. Given a sequence $(\mathbf{x_0}, \mathbf{x_1}, \cdots, \mathbf{x_n})$, each vector $\mathbf{x_t}$ of this sequence is iteratively fed to the LSTM as an input, which produces a hidden state $\mathbf{h_t}$ as well as a cell state $\mathbf{c_t}$. $\mathbf{h_t}$ is used as the output of the LSTM for step t, but the contents of $\mathbf{h_t}$ and $\mathbf{c_t}$ are also used, together with $\mathbf{x_{t+1}}$, as the input to the LSTM for step $t + 1$. The network can therefore forward information to itself in the future (i.e., it can “remember” information). Figure 3 shows how such an LSTM-network is applied to a sequence of inputs.

In practice, the LSTM is not iteratively applied to the sequence, but the iterations are unrolled. During unrolling, for a sequence of length n, the same LSTM is replicated for each of the n iterations, transforming recurrent connections to feed-forward connections, and the resulting bigger system is treated as a single neural network, which can consume the whole sequence at once (see the right side of Figure 3).

What information is encoded in $\mathbf{c_t}$ and $\mathbf{h_t}$ is not predefined but is learned from the training data by the LSTM via multiple internal neural networks. The unrolled network is trained like any other neural network using a loss function and gradient descent. That is, an expected output sequence $(\mathbf{y_0}, \mathbf{y_1}, \cdots, \mathbf{y_n})$ is compared to the actual output of the LSTM $(\mathbf{h_0}, \mathbf{h_1}, \cdots, \mathbf{h_n})$ via an appropriate loss function, a gradient with respect to the network weights is calculated, and gradient descent is used to change the weights of the LSTM in the right direction. Note that all copies of the LSTM that were "produced" during unrolling are still the same network and have to stay identical also during/after training. Therefore, the weight updates of all instances of the LSTM are aggregated and applied to all instances of the LSTM. Since after unrolling, the gradient propagates through all the duplicates of the LSTM for all the elements of the sequence, the LSTM can "learn" to remember some information because it will be helpful later.

Often, the output needed from an LSTM is not a sequence of vectors but a single vector (e.g., for classifying a sequence), in which case only the output $\mathbf{h_n}$ for the last element in the sequence is compared to an expected output, and all the other hidden states $(\mathbf{h_0}, \mathbf{h_1}, \cdots, \mathbf{h_{n-1}})$ are ignored for the loss.

LSTMs and RNNs, in general, have three advantages over feed-forward networks: (1) They can operate over sequences of arbitrary length because the unrolling can be done dynamically. Imagine we want to classify sentences: We can interpret the sentence as a sequence of symbols that we can feed to an LSTM and use the final output of the LSTM to classify some property of the sentence (e.g., its sentiment). Since we can unroll the LSTM to any length we want, we are not restricted by the length of the sentence. At least not in theory; in practice, using an LSTM for much shorter/longer sequences than it was trained on might lead to diminished performance. (2) The fact that the same neural networks process each element of the sequence in the LSTM means that the network can generalize across positions in the sequence (like a CNN can generalize across positions on the two dimensions of an image). For example, if we have to put different panels from a Raven's Progressive Matrix (RPM) test (see Figure 6) into relation to each other, it is intuitive that features extracted for the upper left panel are probably also going to be helpful for the lower right panel and so on. (3) Through the structure of a sequence, we implicitly model that all elements of the sequence are closely related to each other (e.g., all symbols of a sentence, or all panels from an RPM in our case) and most of the relevant information can be inferred by putting them in relation to each other (e.g., the individual symbols in a sentence only really become informative once they are seen as words etc.), which is helpful if we want to learn relational concepts. One problem with LSTMs when modeling relational concepts is that the entities to be put into relation with each other already have to be separated to feed them into the LSTM as a sequence. This splitting does work for many synthetic datasets, but for real images, the entities first have to be separated, which needs some form of attention, supporting Hypothesis 1.

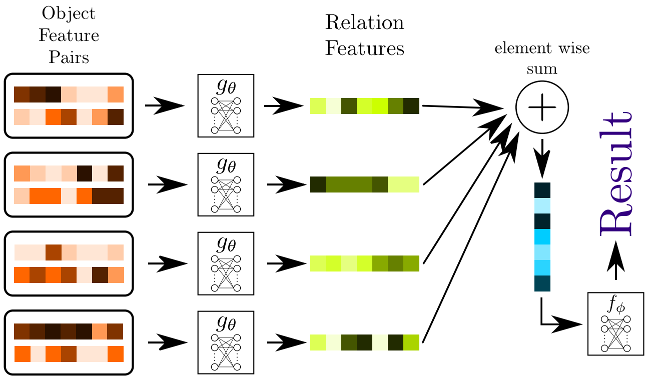


Figure 4. Schematic visualization of a *relation network*. Features of object pairs are sent through the same neural network g_θ, which extracts features encoding the relationship between the objects in each pair. These relation features are added to accumulate the relational information between all object pairs, and the resulting vector is interpreted by a neural network f_ϕ to solve a specific task like classification.

Relation networks (RNs; see Figure 4), introduced by Santoro et al. (2017), are based on the principle of applying a neural network g_θ to all possible "object" pairings to detect relationships between them. The big advantage of this is that the application of g_θ on the object pairs can be done iteratively. Therefore, the network size does not increase with the number of objects to be compared, similar to how the size of an LSTM does not increase with the length of the sequence to be processed. Objects, in this case, are simply features for which a relationship should be detected. The output of g_θ for all pairs is added to integrate the information of possible relationships between all object pairs, and the result is sent through an additional neural network f_ϕ to produce a final classification. This network architecture was able to achieve superhuman performance on the Compositional Language and Elementary Visual Reasoning (CLEVR) dataset by Johnson et al. (2017), which consists of rendered scenes containing different simple objects of varying sizes, colors, and materials (see Figure 5). The dataset also includes written questions that, in part, require relational reasoning to be solved (e.g., "Are there any

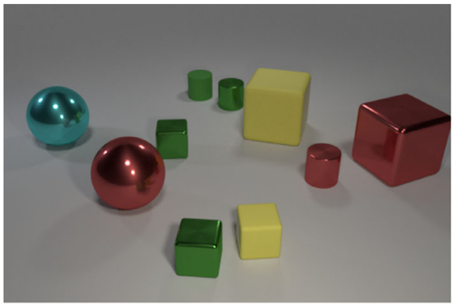


Figure 5. Example image from the CLEVR dataset by Johnson et al. (2017). A possible question for this image could be: “What size does the cylinder with the same color as one of the spheres have? with the correct answer being “small.”

rubber things that have the same size as the blue metallic sphere?”).

In our opinion, the RN architecture has two main bottlenecks: First, given n objects to be compared, g_θ has to be evaluated $\binom{n}{2}$ times, so the number of evaluations of g_θ grows following $O(n^2)$. If relationships between more than two objects should be handled, the number of needed evaluations proliferates. For relationships between r objects, the network g_θ has to be evaluated $\binom{n}{r}$ times, so the number of evaluations grows with $O(n^r)$. Therefore, this approach is only practical if the number of “objects” can be kept relatively small. Without an attention mechanism, Santoro et al. (2017) were not able to directly extract features of objects because there was no information about what part of an image is an object. This is the same problem we already mentioned for LSTMs and supports Hypothesis 1, which states that some form of attention is an essential component of a system to learn relational concepts. The authors decided to extract features from all positions on a grid over the whole image and handle each position as an object. This method, lacking attention, means that the number of “objects” to be compared grows quadratically with the image’s resolution. Also, this increase in object pairs results in more and more relation features that have to be integrated, increasing the likelihood that irrelevant relationships between other object pairs wash out helpful information. Second, given two object features, the network g_θ has to recognize the relationship from the information contained in those features alone. If the relationship to be detected is “similarity,” the representations have to contain all the information to reconstruct the object from it. With more complex objects, these features will become very complex, and a large amount of information must be passed along to f_ϕ. This bottleneck could be circumvented by iterative processing since the comparison could be made in multiple iterations, and in each iteration, only a tiny part of the whole information from both entities has to be compared.

Although the results of RNs on the CLEVR dataset seem pretty promising, the actual variance encoded in a scene is surprisingly small. There are only 96 different combinations of shape, size, material, and color. In essence, this means an object in the CLEVR dataset only contains fewer than 7 bits of relevant information. Some form of positional information, putting the objects in spatial relation to each other, is also needed to solve some of the questions (e.g., “left of,” “behind”) contained in the dataset. Still, this will likely not increase the amount of information needed to encode a complete scene by a considerable amount.

Therefore, it is not clear how well the results of RNs on the CLEVR dataset transfer to real-world tasks. Results with different datasets, which will be presented over the rest of this article, indicate that the performance of RNs decreases for more complex datasets.

Current research on deep learning for visual relational concepts

Since most of the research on deep learning is concerned with perceptual concept learning and the systems perform very well on these tasks by design, we will not analyze this group of tasks in more detail. Furthermore, to the best of our knowledge, there is no explicit research on deep associative concept learning, and we will therefore not analyze these tasks in more detail either. In our opinion, the most interesting tasks can be found within the area of relational concept learning since these tasks seem to be right at the border between solvable and unsolvable tasks for deep learning methods and are also relevant for many practical applications. As mentioned, we will concentrate on “pure” tasks from this domain.

Work on Raven’s Progressive Matrices

Raven’s Progressive Matrices (RPMs), first presented by Raven et al. (1938), are a widely used set of problems to evaluate abstract reasoning and fluid intelligence in humans. Raven’s Progressive Matrices consist of a matrix of abstract images related to each other along the columns or rows following specific rules. One of the images is left blank and has to be selected from a set of candidates to relate to the other images following the established rules. Following Occam’s razor (Schaffer, 2015), the most straightforward rules that can explain the relationships between the images are the correct ones. Figure 6 shows an example of such an RPM.

Learning rules on how a system changes over time and using those rules to predict the future state is, of course, a fundamental property an agent interacting

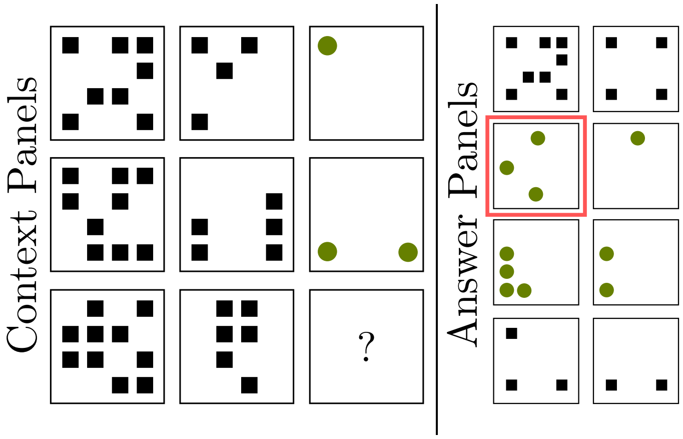


Figure 6. Example of a Raven's Progressive Matrix. The defining property of this RPM is that the number of shapes increases by one from image to image along the columns while preserving the properties of the shapes. The correct solution therefore is the second image of the first column. Adapted from Barrett et al. (2018).

with the world has to master. To learn the rules that drive the change of a system, one has to determine how already observed states relate to each other, and RPMs, in essence, test this capability. This is indicative of how important it is to be able to learn relational concepts.

Collections of RPMs used to test humans are not well suited for machine learning since the number of available examples is usually insufficient. Thus, it would not be possible to distinguish the inherent shortcomings of a method from a simple lack of sufficient training data. Wang and Su (2015) were the first to use an algorithm to generate an arbitrary number of RPMs. This dataset would have been suited for experiments with machine learning systems, but no such experiments have been performed to our knowledge. Fortunately, multiple datasets have been created by now that follow the basic concept of Raven's Progressive Matrices and are specifically designed for machine learning research.

Deep learning and Raven's Progressive Matrices

As far as we can tell, the earliest such work is by Hoshen and Werman (2017), who looked at the performance of neural networks when tasked with choosing or generating the correct continuation of a sequence of changing images, reminiscent of Raven's Progressive Matrices. The networks had to either choose from a predefined set of images (*multiple-choice* task) or had to generate the next image in the sequence directly (*open question* task). Different transformations (e.g., rotation, size, reflection, color) were used to generate the image sequences.

For the *multiple-choice* part, a sequence of images is presented to the neural network, together with a set of possible candidates for the next image in the sequence. The network's task is to select the image that continues the underlying pattern. Figure 7 shows one example of the multiple-choice task. It was solved by the authors using a network architecture similar to AlexNet (a conventional CNN architecture without skip connections) by Krizhevsky et al. (2012), which was used without pretraining on another dataset first. The image sequence and possible solution images were presented to the network as a stack of separate images. Thus, the system did not have to detect and separate the entities and possible solutions independently. The system was able to solve this task with an average accuracy of 97%.

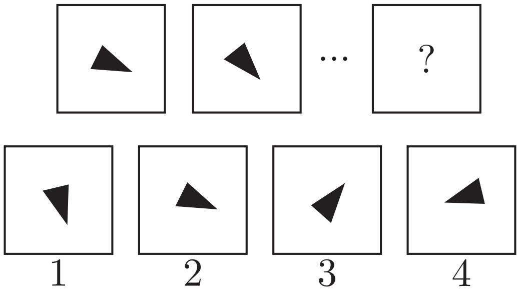


Figure 7. An example for the kind of problems used in the multiple-choice task by Hoshen and Werman (2017). The first two images are given, showing a triangle that is rotated by a constant angle between the first and second image. Four possible continuations of this sequence are given, with Option 1 being the correct one in this case.

For the *open question* part, the network did not select an image from a set of possible solutions but generated the next image directly. The network architecture for these problems was based on the DC-GAN architecture by Radford, Metz, and Chintala (2015), which was also used without pretraining. The performance was measured using the mean squared distance between the ground truth image and the generated image, and the results were also checked qualitatively. The network achieved an average mean squared error of $3.96 \cdot 10^{-4}$, and the resulting images looked qualitatively close to the correct solution.

The results show that even simple CNN architectures are surprisingly good at solving these supposedly complex relational reasoning tasks. Unfortunately, since the networks were trained using 100,000 images and it is hard to judge the actual variability of the dataset, the achieved accuracy could also be the result of memorization by the network.

As previously mentioned, the images were fed to the network as already separated entities, which is *equivalent to an external attention mechanism*. Following Hypothesis 1, this already removes one of the main difficulties of such a task. Kim, Ricci, and Serre (2018b) also pointed this out in a different context. In our opinion, the dataset is therefore not suited for judging

Model	Accuracy
Blind ResNet	22%
CNN	33%
LSTM	36%
ResNet-50	42%
Wild-ResNet	48%
WReN	63%

Table 1. Average accuracy of different architectures tested by Barrett et al. (2018) on the Procedurally Generated Matrices dataset. Adapted from the original article.

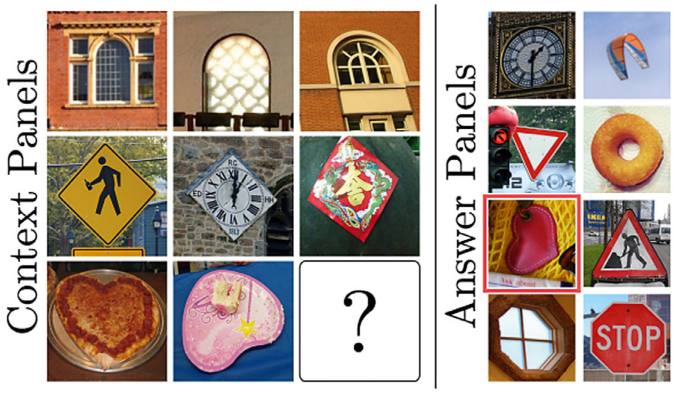


Figure 8. Example from the V-PROM dataset by Teney et al. (2019). In this example, the images in the Context Panels are related to each other along the rows by their shape. One image is left blank, and the correct image, the heart shape, has to be selected from the Answer Panels. Adapted from the original article.

a system under real-world circumstances, where such a preattention mechanism usually is not present.

Procedurally generated matrices

Barrett et al. (2018) extended on the ideas by Hoshen and Werman (2017) and replicated RPMs more closely. The authors call this dataset the *Procedurally Generated Matrices* (PGM) dataset, which is freely available. Figure 6 shows a visualization of an example from this dataset. Different architectures were trained and tested on the PGM dataset. The data were again provided to the network as an image stack of 16 separate images (the eight context panels and the eight answer panels). The networks had to select the right panel from the provided answer panels. The rules used for generating the RPMs are pretty elaborate, and we would like to refer the reader to the original article for more information.

Five different network architectures were tested: (1) a *simple CNN*; (2) a more modern CNN architecture utilizing skip connections in the form of a *ResNet-50* by He et al. (2016); (3) an *LSTM* based on a variant by Zaremba, Sutskever, and Vinyals (2014) together with a small CNN for feature extraction; (4) a novel adaptation of a relation network (Santoro et al., 2017), which the authors named a *Wild relation network* (WReN) for which multiple relation networks work in parallel; and (5) an adaptation of ResNet, which the authors named *Wild-ResNet* for which a ResNet-50 is separately evaluated for each answer panel. A second version of the ResNet architecture, which the authors named *context-blind ResNet*, was used to detect unwanted statistical regularities in the dataset. The context-blind ResNet was only given access to the answer panels and therefore had to rely purely on statistical properties of the answer set to solve the tasks. In essence, the result from the context-blind ResNet is the baseline accuracy of a system that does not know the question to be answered. All networks were used without pretraining.

The average performance for the whole dataset can be seen in Table 1. The results on the PGM dataset are pretty surprising, considering that the same, simple CNN architecture achieved 97% accuracy for the dataset used by Hoshen and Werman (2017). The CNN only performs slightly better than the blind ResNet, which can be seen as the random baseline accuracy, showing that the dataset by Hoshen and Werman might lack in some way. Either the variability is not large enough in relation to the number of training samples used, which might lead to rote memorization by the network, or the dataset contains statistical correlations that can be used for classification. The WReN architecture performs much better with an accuracy of 63% but is still far from perfect. The research by Barrett et al. (2018) would indicate that CNNs, as well as recurrent neural networks, seem to have difficulty with tasks that require more complex relational reasoning, even if the entities are preattended. Similar to the previous dataset, the fact that the entities between which the relations should be detected are already separated makes it difficult to judge how the results on the PGM dataset would transfer to a real-world scenario. Especially, the best-performing architecture, utilizing relation networks, heavily relies on this preattention, supporting our Hypothesis 1 that attention is vital to solve relation reasoning tasks.

Visual Progressive Matrices

Teney et al. (2019) released a conceptually similar dataset named Visual Progressive Matrices (V-PROM) using natural instead of synthetically generated images. See Figure 8 for an example. The authors also include a wide variety of carefully crafted training/testing splits of the dataset to evaluate the generalizability of systems for specific concepts. There are, among others, sets to test how well a system generalizes the concept of counting to unseen numbers and to test if the system

	ResNet	ResNet + aux.loss	B.-up	B.-up + aux.loss
Human accuracy	78%	78%	78%	78%
RN, shuffled input (base accuracy)	13%	13%	13%	13%
MLP	41%	45%	50%	56%
GRU	43%	48%	46%	53%
Top-VQA	37%	40%	38%	41%
Relation Network (RN)	**51%**	**56%**	**55%**	**61%**

Table 2. Accuracy over the whole V-PROM Dataset from Teney et al. (2019). Adapted from the original article.

generalizes to unseen object categories. The dataset was tested on different network architectures. First, features were extracted from the images using one of two pretrained CNNs (either a ResNet101 by He et al. [2016] or a Bottom-Up Attention Network by Anderson et al. [2018]). These extracted features were then interpreted by either a simple multilayer perceptron, a recurrent neural network using gated recurrent units by Cho et al. (2014) (a simplified version of an LSTM), the current top-performing method for visual question answering (Teney, Anderson, He, & Hengel, 2018), and a relation network by Santoro et al. (2017). All systems were trained to either select the correct image or explicitly classify the relationships underlying the images as an auxiliary loss. This auxiliary loss was only used during training to guide the networks to learn good internal representations and proved very useful. In some sense, this loss can be seen as giving additional information to the system about what task it is currently learning. The authors showed that even the best-tested system (again a relation network) was not able to approach human performance (see Table 2).

Similar to the datasets by Hoshen and Werman (2017) and Barrett et al. (2018), the images of the V-PROM dataset were provided to the tested system in an already preattended, separated way. This again makes it difficult to judge how well the experimental results would transfer to the real world. Considering that relation networks, again the best-performing architecture, profit highly from this preattended form of data strengthens Hypothesis 1, that an attentional mechanism will be an essential component in a system that can solve relational reasoning tasks.

The SVRT dataset

The SVRT dataset (Fleuret et al., 2011) was created to test the abstract reasoning capability of computer vision systems and compare it to human performance. The dataset consists of 23 problems that are trained for and tested independently. The goal for all the problems is to categorize images (showing abstract shapes) into one of two classes that are separated by some abstract property. For example, in Problem 1 (see Figure 9 for example images), two shapes are present. For Class 1, the shapes are different, and for Class 2, they are identical. Being able to detect similarity is an essential task for any intelligent system to perform searches and identify out of place/novel objects. The SVRT dataset relies heavily on the ability to detect similarity for many of the problems.

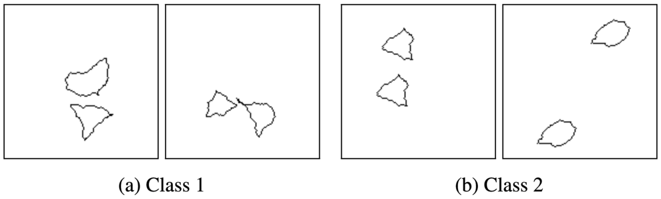


Figure 9. Example images for Problem 1 of the SVRT dataset by Fleuret et al. (2011).

When the SVRT dataset was created, deep learning was not yet mainstream, so the authors did not test the dataset on those methods. The best-performing method tested by Fleuret et al. was Adaboost by Freund and Schapire (1997), using the Feature Group 3, which includes the “number of black pixels in a rectangular subregion of the image for a large number of such regions[,]... information about the distribution of edges[,]... spectral properties of the image (Fourier and wavelet coefficients)” (Fleuret et al., 2011). Using a Fourier transform was also recently used to solve the SVRT dataset by Bohn, Hu, and Ling (2019).

Before delving deeper into research done on the SVRT dataset, we would like to mention one potential flaw this dataset might have, in our opinion. A random process is used to generate the shapes for the images. If identical shapes are required, one randomly generated shape is copied pixel by pixel and recaled and rotated if necessary. If different shapes are needed, the random shape generation process is used multiple times. This way of producing images means that shapes that should be considered identical are identical (up to scaling and rotation for some of the problems), and shapes that should be considered different are most likely not even roughly resembling each other (see Figure 9). Thus, in most cases, two shapes that approximately resemble each other will be identical. Therefore, it might be

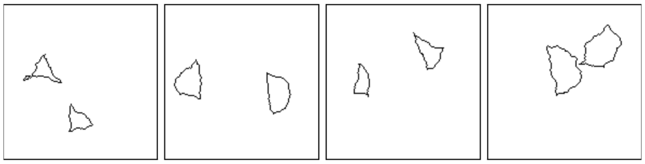


Figure 10. Examples of incorrectly classified images of the "different-class" from Problem 1 of the SVRT dataset. A ResNet-50 network was trained on 28,000 images and the presented images were misclassified.

enough for a system to detect a few rough local features to be reasonably sure that two shapes are identical without really comparing them. Looking at images misclassified by a well-performing neural network, it is easy to imagine that a network that uses local features to detect similarity might think the presented shapes are identical. See Figure 10 for images that are misclassified by a ResNet-50 architecture with above 90% accuracy on Problem 1, where the goal is to detect whether two shapes in an image are identical or not. For example, the first image shows two shapes with three sharp corners, the second two sharp corners and one bigger "arch," and so on. In addition, the relations to be learned in the SVRT dataset are relatively simple and mainly consist of recognizing similarity and the spatial orientation shapes. The performance on the SVRT dataset might therefore overestimate a system's ability to learn relational concepts truly.

We started to evaluate deep learning methods on the SVRT dataset in Stabinger, Rodríguez-Sánchez, and Piater (2016b) and greatly extended those experiments in Stabinger, Rodríguez-Sánchez, and Piater (2016a) by testing how well an old (LeNet by LeCun et al., 1989) and a new (GoogLeNet by Szegedy et al., 2015) CNN architecture performed on the SVRT dataset. We trained LeNet and GoogLeNet for each problem, except for Problems 3, 11, and 13, for which we could not generate images with the required size. The models were trained separately for each problem with 40,000 images and tested using 20,000 images. The LeNet architecture was trained from scratch, and the GoogLeNet architecture was pretrained on ImageNet. Even though the presented images look very different from natural images, we found that using a pretrained network on natural images converges faster during training.

One other goal was to compare the performance of CNNs to that of humans who were tested by Fleuret et al. (2011). Unfortunately, this is not directly possible since individual subjects in practice either achieve 100% accuracy on a problem if they can figure out the underlying rule separating the classes or achieve accuracy close to chance if they are not able to figure out the rule. Therefore, we report the percentage of human subjects that were able to solve the given problem.

Table 3 shows the results for both tested network architectures, in addition to all other results on this dataset by research presented in this article at a later point.

A few surprising facts emerge: First, the best method by Fleuret et al. (2011) outperforms many of the more modern architectures on average. Second, the more modern GoogLeNet architecture performs slightly worse than the much older and simpler LeNet architecture. Third, and most interestingly, there seems to be a prominent grouping of problems around the concept of shape comparison. The other results will be discussed later in this article.

Problems for which the shapes of the entities are related to each other (*same–different* problems) are complex for CNNs and problems where the positions of the entities stand in specific relation to each other (*spatial–relation* problems) are easy for CNNs. This is especially evident when looking at a graphical visualization of the achieved accuracies (see Figure 11). For the *spatial–relation* problems, LeNet and GoogLeNet perform better than the best method used by Fleuret et al. (2011). In addition, the newer GoogLeNet performs significantly better than the old LeNet architecture, almost reaching an average accuracy of 100%. For the *same–different* problems, the performance of the CNN architectures is much worse. Both architectures do not achieve an accuracy that is significantly above chance in almost all of the cases.

Three problems seem to go against the general trend (namely, Problems 6, 16, and 17). A system should theoretically need to perform shape comparison to solve these problems, but we could show that additional information in the dataset enabled the CNNs to correctly classify the images without the need to compare shapes (see Stabinger et al., 2016a, for a more in-depth explanation). This demonstrates that one has to take great care when creating a dataset to test the performance of machine learning systems since they readily exploit properties of the dataset one did not intend to be used for classification.

Ricci et al. (2018b) independently performed very similar experiments to us on the SVRT dataset. The authors also tested convolutional neural network architectures on this dataset but used a whole set of CNNs to check whether the performance difference of same–different tasks from spatial–relation tasks was influenced by the architecture. We refer the reader to the original article for a detailed description of the used architectures and training procedures.

Ricci et al. (2018b) confirmed the finding by us that CNNs seem to be particularly challenged by tasks that require the comparison of "objects." The authors could also show that the size of the network was less

Problem	LeNet[a]	GoogLeNet[a]	Small CNNs[b]	CorNet-S[c]	ResNet-50[d]	Fourier[e]	Adaboost[f]		
Parameters	60,850	7 Mio	Few 10k (varied)	106 Mio	23 Mio	138 Mio	Unknown		
# Images	20k	20k	1 Mio	400k	100/1k/28k	20	10k	Human[g]	
Pretrained	No	ImageNet	No	ImageNet	ImageNet	ImageNet	No	Average 6.3	Rule
1	57%	50%	62%	100%	59%/88%/100%	100%	98%	98%	Compare
5	54%	50%	67%	97%	56%/69%/99%	96%	87%	90%	Compare & grouping
6	76%	86%	86%		58%/71%/99%	51%	76%	70%	Compare & grouping
7	53%	50%	57%		56%/61%/100%	61%	76%	90%	Compare & grouping
15	52%	50%	68%		86%/100%/100%	100%	100%	95%	Compare
16	98%	50%	76%		84%/100%/100%	99%	100%	78%	Compare
17	75%	95%	88%		69%/83%/97%	53%	67%	78%	Compare & position
19	51%	50%	60%		54%/76%/99%	57%	61%	98%	Compare
20	55%	50%	56%	95%	52%/56%/93%	56%	70%	98%	Compare
21	51%	51%	59%	96%	51%/70%/99%	51%	50%	83%	Compare
22	59%	50%	63%		70%/97%/100%	98%	97%	100%	Compare
2	100%	100%	100%		100%/100%/100%	78%	98%	100%	Position
3			100%		95%/100%/100%	58%	95%	100%	Position
4	98%	100%	100%		100%/100%/100%	67%	93%	100%	Position
8	94%	91%	95%		92%/99%/100%	83%	90%	100%	Position
9	93%	100%	89%		81%/96%/96%	51%	68%	93%	Size & position
10	99%	100%	100%		97%/100%/100%	84%	94%	98%	Position
11			100%		100%/100%/100%	64%	96%	100%	Position
12	97%	100%	100%		94%/100%/100%	57%	84%	95%	Size & position
13			91%		63%/97%/100%	70%	67%	93%	Position
14	90%	100%	97%		73%/99%/100%	68%	73%	98%	Alignment
18	99%	99%	100%		92%/99%/100%	54%	99%	93%	Grouping
23	87%	100%	94%		95%/100%/100%	55%	75%	100%	Position
Average	77%	76%	83%	97%	77%/90%/99%	70%	83%	93%	

Table 3. Aggregation of results for the SVRT dataset by Fleuret et al. (2011). The two groups indicate problems that entail same–different relations or not.

[a] Stabinger et al. (2016a).

[b] Results of the best-performing CNNs per problem by Ricci et al. (2018b) (reconstructed from the published graph).

[c] Messina et al. (2019).

[d] Results with 100/1,000/28,000 training images by Funke et al. (2019) (reconstructed from the published graph).

[e] Best results per problem by Bohn et al. (2019).

[f] Boosting with Feature Group 3 by Fleuret et al. (2011).

[g] Human accuracies as estimated in Stabinger et al. (2016a) using experimental data by Fleuret et al. (2011).

critical for the spatial–relation problems of SVRT (i.e., problems where the positioning of shapes is essential) in comparison to the same–different problems (i.e., problems that rely on the comparison of shapes). The overall performance reported by Ricci et al. (2018b) (see Figure 12 and Table 3) is higher than what we were able to achieve in Stabinger et al. (2016a). This difference in performance is likely a result of using more images for training. Ricci et al. (2018b) also put some of the problems in the opposing group, but these differences do not change the overall conclusion.

The main conclusion from these experiments is that convolutional neural networks have greater difficulty detecting same–different relations than they do to detect spatial relations. This dichotomy could be explained by spatial relations more closely resembling a perceptual concept since the global arrangement of objects can often be solved by simple pattern matching, whereas same–different problems are a classic example of a relational concept. This strengthens our Hypothesis 2, that current neural network architectures have more significant problems with learning relational concepts than learning other concepts.

It is also noteworthy that neither the method presented by Fleuret et al. (2011) nor the human experiments show a clear difference between the two groups of problems, so the learning of same–different relations does not seem to be more difficult in general but is especially challenging for convolutional neural networks.

Solving the SVRT dataset

In 2019, Messina et al. (2019) were able to solve Problems 1, 5, 20, and 21 of the SVRT dataset. The authors were able to achieve an accuracy of above 95% for all four problems using different ResNet architectures by He et al. (2016) as well as the

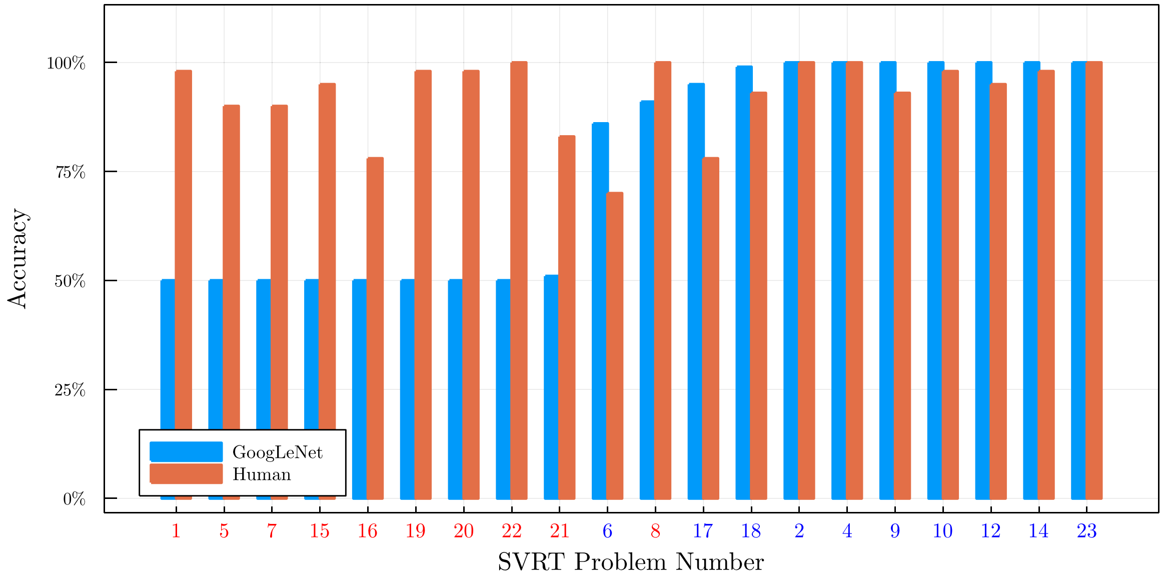


Figure 11. Graphical visualization of the accuracy of GoogLeNet and of humans as reported in Stabinger et al. (2016a) on the problems of the SVRT dataset by Fleuret et al. (2011). Same–different problems have a red number and spatial–relation problems a blue one.

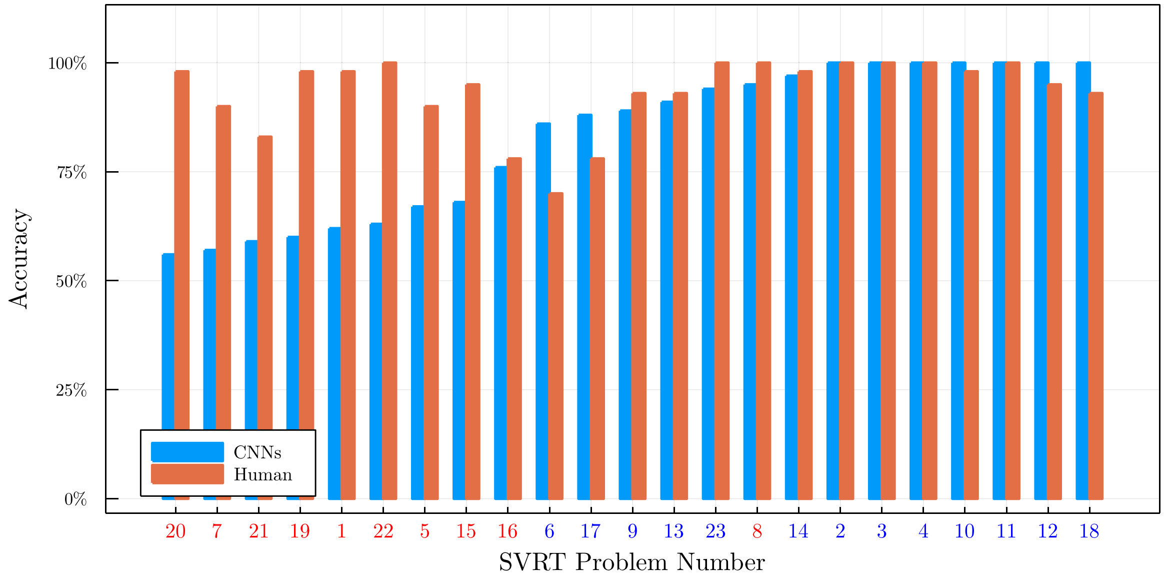


Figure 12. Accuracy achieved by Ricci et al. (2018b) and by humans in Stabinger et al. (2016a) for the problems of the SVRT dataset by Fleuret et al. (2011). Same–different problems have a red number and spatial–relation problems a blue one. Compare with Figure 11.

biologically inspired CorNet-S architecture by Kubilius et al. (2018) (see Table 3 for the CorNet-S results), but the authors had to use 400,000 training images to achieve these results, which might be problematic, considering the discussed potential problems of the SVRT dataset. Both networks were pretrained on the ImageNet dataset before being fine-tuned for the actual task.

Bohn et al. (2019) were also able to solve many of the same–different tasks of the SVRT dataset using deep learning while only utilizing 20 training images (see Table 3). They were able to achieve this by extracting the amplitude spectrum of the Fourier transform of the images. As the authors note, it is well known that peaks in the amplitude spectrum correspond to periodic patterns in the image. The peaks, therefore, encode similarity information in a much easier to use form for machine learning methods. Since the difficult part of the task (finding similarities) was more or less solved in a preprocessing step and not by the neural networks, these results do not change our general conclusions about the performance of neural networks for relational concept learning. Nevertheless, it might be a good idea to add this preprocessing step to systems that have to deal with same–different relations in real-world settings. It should be noted that Fleuret et al. (2011) already recognized the importance of spectral data since Fourier and wavelet coefficients were already part of the features used in the original SVRT study and might explain the strong performance of the original method.

Funke et al. (2019) were finally able to achieve accuracies above 90% for all problems of the SVRT dataset (see Table 3) without using special preprocessing steps and with a more reasonable amount of 28,000 training images, using a ResNet-50 architecture that was pretrained on ImageNet. Our Hypothesis 1, that relational concept learning is more difficult for current neural network architectures than other concepts, is still valid, since the results when training on 100 and 1,000 images still exhibit a big difference between same–different problems (which is a relational concept) and spatial–relation problems (which are closer to perceptual concepts). In addition, our experiments in Stabinger, Peer, and Rodríguez-Sánchez (2021) on a much more tightly controlled same–different task (inspired by the PSVRT dataset, presented in a later section) showed that ResNet does not perform well on same–different tasks in general. This might indicate that the good results by Funke et al. (2019) are more indicative of the previously discussed shortcomings of the SVRT dataset than the suitability of ResNet architectures for solving relational concept learning.

The Bongard problems

The SVRT dataset is somewhat reminiscent of the problems presented by Bongard (1970) as examples of problems a neural network would never be able to solve. It has to be noted that the tasks by Bongard were more difficult than those of the SVRT dataset because the goal was not to classify images but to give a textual description of what separates the two classes. Hofstadter popularized similar problems with his book *Gödel, Escher, Bach: An Eternal Golden Braid* (Hofstadter, 1979). Figure 13 shows an example for such a "Bongard problem." The goal is to describe what abstract property separates the images on the left from those on the right. In the case of Figure 13, the images on the left show convex shapes while the images on the right show concave shapes. To our knowledge, Depeweg, Rothkopf, and Jäkel (2018) are the only researchers in recent years who tried to solve Bongard problems as they were originally intended (i.e., trying to generate an explanation of how two sets of abstract images differ), but they did not use deep learning to do so. Nie et al. (2020) created a dataset, called Bongard LOGO, which is inspired by the Bongard problems. Unfortunately, the dataset does not contain relational tasks, so we will not cover them in this article. Yun et al. (2020) worked on actual Bongard problems using deep learning as part of their system but only tested few-shot classification and also not the original task of generating descriptions. Considering the recent success in image caption generation by, for example, Xu et al. (2015) Donahue et al. (2015), Fang et al. (2015), and many more, the Bongard problems, in their original form, might be an interesting topic for future deep learning research.

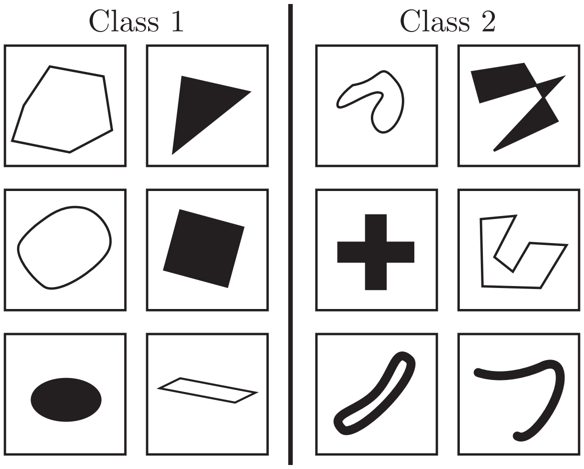
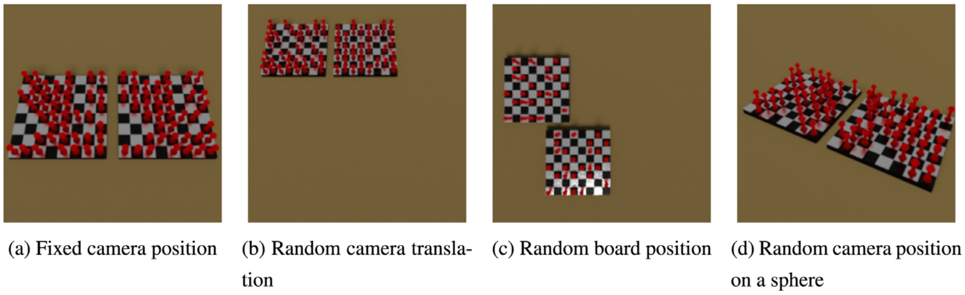


Figure 13. Example of a "Bongard problem." The differentiating property in this case is that the images in Class 1 show convex shapes while the images in Class 2 show concave objects.

The chess dataset

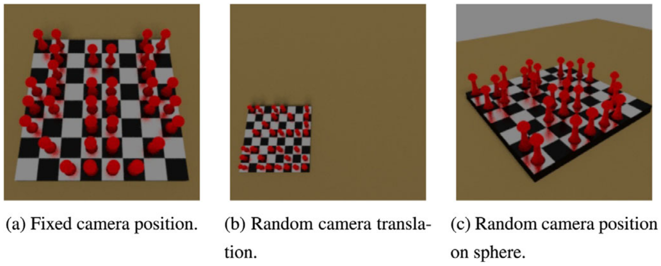
As discussed, the SVRT dataset has its flaws. In Stabinger and Rodríguez-Sánchez (2017), we tried to create a more robust dataset, which also more closely resembles natural images, by rendering them in a seminaturalistic way using a three-dimensional rendering software (Blender Online Community, 2017). Each image shows one or two chessboards with red pawns randomly placed on the field.

The dataset consists of two distinct tasks: The goal of the *identity task*, showing two chessboards in each image, is to detect whether the pawn positions are identical on both boards. The goal for the *symmetry task*, showing one chessboard in the images, is to detect whether the pawn placement is symmetric. The difficulty of both tasks was controlled by allowing translation of the chessboards and the camera, movement of the camera on a virtual half-sphere around the chessboards, and a varying amount of pawns that break the identity/symmetry property. Example images can be seen in Figures 14 and 15.



(a) Fixed camera position (b) Random camera translation (c) Random board position (d) Random camera position on a sphere

Figure 14. Variations of the identity task in the chess dataset.



(a) Fixed camera position. (b) Random camera translation. (c) Random camera position on sphere.

Figure 15. Variations of the symmetry task in the chess dataset.

We trained AlexNet by Krizhevsky et al. (2012), VGG16 by Simonyan and Zisserman (2015), and GoogLeNet by Szegedy et al. (2015) on all variations of the two tasks with 1, 5, and 10 out-of-place pawns. All networks were pretrained on the ImageNet dataset by Deng et al. (2009), after which the last layer was replaced by a new, randomly initialized layer, to conform to the smaller number of classes of the chess dataset (from 1,000 classes down to 2), and the whole network was trained on the chess dataset, without fixing any of the pretrained layers.

In addition, we had to employ a training scheme that is related to curriculum learning, first proposed by Bengio, Louradour, Collobert, and Weston (2009). To learn the more complicated variants of a task, we started from networks already successfully trained on easier variations of the same task. Without employing this training scheme, we could not train networks to solve the tasks with only one pawn that breaks symmetry/identity.

The results showed that the symmetry task is considerably more manageable than the identity task, but both tasks cannot be learned in all cases. For example, GoogLeNet was not able to achieve accuracies significantly above chance for the most challenging task (i.e., camera rotation with only one out-of-place pawn), and for the identity task, the network was not able to perform better than chance on any of the tasks with camera rotation, which supports Hypothesis 2.

Still, the results on the identity task were surprisingly good, considering that GoogLeNet is not able to solve the simple task of comparing two shapes from the SVRT dataset and the chess dataset, with random placement of the checkerboards seeming much more complicated but can be solved quite well by GoogLeNet.

One explanation might be that our curriculum learning approach might be very helpful for such abstract tasks. Unfortunately, it is not immediately clear how to transfer curriculum learning to the SVRT or other datasets, or even more real-world scenarios, because the difficulty of the produced samples cannot easily be controlled. In addition, the very regular, never-changing, and easy to detect grid of the checkerboard might help the network extract the needed pawn positions, despite the high variability of the images.

The parametric SVRT dataset

Similar to our reasoning for the chess dataset Ricci et al. (2018b) recognized that the generation procedures for the SVRT dataset are too unpredictable to lead to reliable conclusions. The authors specifically mention that it is sometimes unclear whether a problem cannot be solved because of the relations the network has to learn or because the variability of the images (i.e., the size and number of shapes) has increased. To further investigate the findings from the SVRT dataset that CNNs are better at learning spatial relations than they are at learning same–different relations, the authors did a second experiment where they created their own, simple dataset. They call this the *parametric SVRT (PSVRT)* dataset. In the PSVRT dataset, each image contains two patches, composed of black and white boxes, on a neutral background. Examples for this dataset can be seen in Figure 16. The two patches have two properties that can be used for classification. The first is the same–different relation depending on whether the two patches show the same black and white pattern. The second property is the spatial positioning of the patches, depending on whether the two patches are oriented horizontally or vertically to each other. Three parameters control the amount of variability

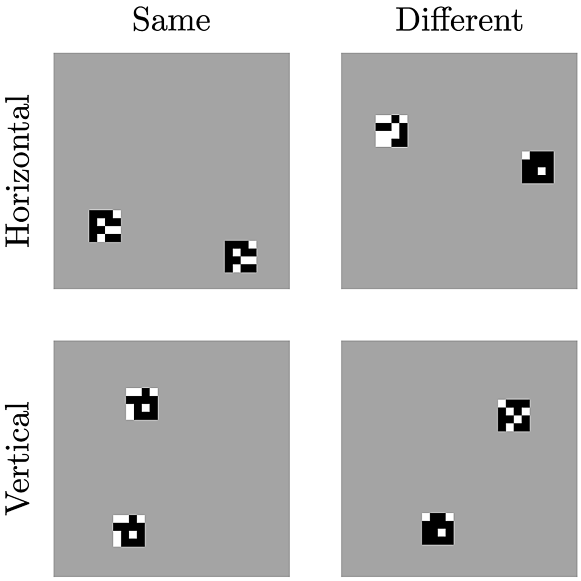


Figure 16. Examples for all four class combinations of the PSVRT dataset presented by Ricci et al. (2018b). An image can be *same* or *different*, depending on whether the two patches show the same pattern, and *horizontal* or *vertical*, depending on the orientation of the two patches. Adapted from the original article.

in the images: the size of the patches, the number of patches, and the image size. Setting up the dataset in this way allows a system to learn the same–different as well as the spatial–relation problem with identical images and ensures that the image complexity and variability are constant between the two problem sets. We refer the reader to the original study for further implementation details.

Using PSVRT, the authors were able to show a sharp dichotomy between solving spatial–relation and same–different tasks as well. The networks consistently learned the classification early in the training procedure for spatial–relation tasks and achieved high final accuracy. For same–different tasks, the performance was highly dependent on the image size. Bigger image sizes led to slower training and lower-end accuracy and resulted in the networks failing more often at learning the task at all, depending on the random seed used for initialization of the network. In addition, the size of the network (the number of parameters) did not influence the achievable accuracy much for spatial–relation tasks but did so for same–different tasks. Since the same images were used in both experiments, the authors conclude that image variability was not what hindered CNNs, and learning the same–different relation problem itself is what is more difficult for the networks, supporting Hypothesis 2 that relational concept learning is more difficult for current neural network architectures than learning other concepts.

The authors hypothesize that the network learns subtraction templates to solve the same–different task because the image's patch size and number do not seem to influence the achievable accuracy. The authors argue that more subtraction templates would only be needed if the number of possible patch positions changes, increasing exponentially with growing image size. Unfortunately, the authors do not provide a more detailed explanation for their hypothesis.

The PSVRT dataset has one unfortunate flaw: The patches are not matched for the number of black and white pixels if they are different. Therefore, a simple comparison of the sum of all pixel values between different image parts is sufficient to "compare" the patches in many cases. If this is what the authors mean by subtraction templates, then we agree that the networks might use this, but we would argue that this is a flaw of the dataset and not an explanation of how comparison, in general, could be solved by a CNN. Future experiments should test the PSVRT dataset with patches with a matched number of black and white pixels.

Relation networks and siamese networks applied to the PSVRT dataset

Kim et al. (2018b) extended the work by Ricci et al. (2018b) by testing the PSVRT dataset with two additional network architectures. The first was an RN proposed by Santoro et al. (2017), which was specifically designed to learn relationships between objects. Kim et al. (2018b) hypothesize that the original performance of the architecture on the CLEVR dataset mainly stems from memorization since, as previously mentioned, the dataset only has a minimal amount of variation. The authors could support this hypothesis by testing RNs on the PSVRT dataset and showing that performance decreases with image size in the same way for relational networks as it does for CNNs until the architecture can not learn the task at all at an image size of 180 × 180 pixels. As previously argued, we think that the increased number of relation features that have to be integrated might pose another problem. Of course, this could be circumvented using attention, strengthening Hypothesis 1. Also, with increasing patch size, it might become challenging to pass all needed information to the network that integrates all relationships (i.e., f_ϕ).

The second architecture tested by Kim et al. (2018b) is a type of *Siamese network*, first proposed by Bromley, Guyon, LeCun, Säckinger, and Shah (1994). Siamese networks were specifically designed to make same–different decisions for images. The caveat of Siamese networks is that the network expects the objects to be compared as separate images (i.e., preattended) (see Figure 17 for a schematic visualization). As Kim et al. (2018b) point out, this splitting into two images can be interpreted as a kind of attention mechanism

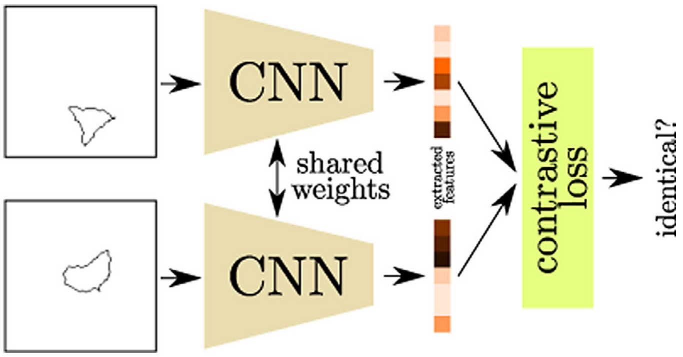


Figure 17. Schematic visualization of a *Siamese network*. Two images to be compared are passed through the same CNN to extract high-level features. These features are then compared using a contrastive loss to determine the similarity of the original images.

simulating the effects of perceptual grouping. The authors were able to show that such Siamese networks can solve PSVRT successfully, not showing a qualitative performance difference between the same–different and the spatial–relation tasks. The performance was also independent of the image parameters (i.e., image size, patch size, and the number of patches). These results are in support of Hypothesis 1, which states that attentional mechanisms are an important component in solving relational concept learning. We could also show in Stabinger et al. (2021) that just separating the entities to be compared into different channels of a normal CNN made a task similar to PSVRT considerably easier.

Discussion and future work

In the beginning, we stated two hypotheses: With Hypothesis 1, we stated that “attentional mechanisms will be an important component to successfully and efficiently learn relational concepts.” We have shown that relation networks, which generally perform very favorably on relational reasoning tasks, need attention to keep the number of object comparison operations low. This is especially important if relations between more than two objects have to be detected. Kim et al. (2018b) as well as our research in Stabinger et al. (2021) show that preattending data considerably improves performance for relation tasks. In addition, as can be seen in Table 4, this preattention is already an integral part of many datasets that are currently used to test systems for learning relational concepts. This demonstrates that many researchers have recognized this need, even if they did not communicate or consider this themselves from the viewpoint of attention. In our opinion, this explains why the results on these datasets are surprisingly good, considering that the straightforward PSVRT dataset can only be solved using massive amounts of training data, and even in these cases, the results are far from perfect. We think that datasets with a form of preattention grossly overestimate the performance, which could be expected under real-world conditions, where such a form of preattention is not available.

One promising group of architectures that integrates attention at its core and has gained traction in many fields of deep learning over the last year are transformer architectures. Vaswani et al. (2017) first proposed these for the field of machine translation; Devlin, Chang, Lee, and Toutanova (2018) later generalized them to many other natural language processing tasks, and by now, they are also heavily researched for many other tasks, including computer vision by Dosovitskiy et al. (2020) and Jaegle et al. (2021), among many others. All transformer architectures contain a self-attention mechanism as one essential building block and, therefore, should be a promising architecture to study for relational reasoning tasks.

With Hypothesis 2, we stated that “relational concepts are more difficult to learn for current neural network architectures than other concepts.” We would argue that despite the progress in recent years, it is still evident that deep learning methods have a weakness in relational reasoning tasks. Results are either not on par with human performance (PGM, V-PROM, PSVRT, chess dataset), might be results of weak datasets (IQ dataset, SVRT, CLEVR), or are unrealistic because the datasets have an attention mechanism embedded in the way the data are presented (IQ dataset, PGM, V-PROM). Our work (Stabinger et al., 2016a) was

Name	Citation	Problems
IQ Dataset	Hoshen and Werman (2017)	Dataset is preattended, unknown variance
Procedurally Generated Matrices	Barrett et al. (2018)	Dataset is preattended
V-PROM	Teney et al. (2019)	Dataset is preattended
SVRT	Fleuret et al. (2011)	Possibly low variance of the images
PSVRT	Kim et al. (2018b)	Can be solved using pixel value sums
CLEVR	Johnson et al. (2017)	Low variance of the scenes
Chess Dataset	Stabinger and Rodríguez-Sánchez (2017)	Static checkerboard might give too many hints

Table 4. Datasets presented in this article and possible problems with them.

the first to show this divergence in performance for different kinds of concept learning. This dichotomy was later also shown by Ricci et al. (2018b) and is currently demonstrated exceptionally well with the PSVRT dataset by Kim et al. (2018b). In our opinion, the recent advances on the SVRT dataset are more indicative of possible shortcomings of the dataset and less of advances of the methods, especially considering the poor performance of the same architectures on the conceptually very similar PSVRT dataset. All of the datasets we presented in this article have, in our opinion, one or more problems (see Table 4).

We think the PSVRT dataset by Ricci, Kim, and Serre (2018a), with an added restriction to pixel value matched patches, to prevent the system from using a simple sum to compare patches, would likely provide the cleanest datasets to test relational reasoning capabilities while minimizing the chance of introducing secondary features a deep learning system can use to “cheat” at the task.

In our opinion, even simple bottom-up attention will not be sufficient to solve relational tasks efficiently, and iterative attention shifts will be necessary to efficiently solve many relational concept learning tasks in the real world. Attention solves the problem of separating entities to be compared but does not solve the problem of information density. For example, if two objects have to be compared for identity, all information about the two objects has to be forwarded to a subsystem that can decide on identity. As the objects’ variability increases, this will likely mean that the layers transporting this information, and the network deciding on identity, will both grow rapidly, making the system inefficient and data-hungry. We theorize that iteratively shifting attention will more favorably balance network size with computation time. We also think that the necessarily shared parameters and substructures will lead to a reduced need for training data and better generalization for relational tasks. This is already partly put into effect in relation networks by the iterative application of g_θ, which might be one of the reasons it performs better on average for relational tasks than most other network architectures.

We think future research should concentrate on creating datasets that test for relational reasoning, without providing a form of preattention or introducing unwanted features that can be used to “cheat” the task. Creating non-preattended instances of existing datasets (like PGM, V-PROM, and the IQ dataset) might be able to further demonstrate the importance of attention for such tasks. Architectures with iterative processing of the input and a mechanism to shift attentional focus between the iterations are currently not very popular but should be investigated more deeply with respect to relational concepts.

Conclusion

In this study, we have summarized and interpreted current deep learning research from the perspective of concept learning. We were able to show that *perceptual concepts* are easily solved by deep learning methods since they were initially developed for this class of problems. *Associative concepts*, even though preliminary evidence suggests that at least some deep learning architectures do implicitly learn such concepts in their intermediary representations, seem to have not been studied until now in the field of deep learning. Thus, we focused our analysis on work that can be classified as learning visual *relational concepts*.

We hope we were able to convince the reader that relational concepts are of practical importance and seem to be particularly difficult for current neural network architectures to learn. We also demonstrated that attentional mechanisms would be, together with a form of an iterative attentional shift, an essential component in solving these problems in the future.

We have also demonstrated that many of the currently used datasets are not ideal and likely overestimate the actual performance of tested systems. Many datasets have a form of preattention built in, or the complexity and variability of the produced samples might be overestimated. New datasets that take these findings into account will therefore have to be created for future research.

Keywords: deep learning, concept learning, relational concepts

Acknowledgments

Commercial relationships: none.
Corresponding author: Sebastian Stabinger.
Email: sebastian@stabinger.name.
Address: Universität Innsbruck, Innsbruck, Austria.

Footnotes

[1] A widely used activation function is the ReLU function $f(x) = \max(0, x)$, which is differentiable for all inputs but 0. In practical implementations, it is common to return, if possible, one of the one-sided derivatives for points that are not fully differentiable. Even though this is mathematically not completely accurate, it works well in practice.

[2] Zentall, Galizio, and Critchfield (2002) provide a more in-depth overview of these three concept classes, and Murphy (2004) gives a comprehensive overview of concepts in general.

[3] For the top-5 error rate, five predictions of the correct class are made, and an image is classified correctly if one of the five predictions is the correct one.

References

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., ... Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6077–6086).

Barrett, D., Santoro, A., Hill, F., Morcos, A., & Lillicrap, T. (2018). Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning* (pp. 4477–4486). PMLR.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 41–48).

Blaisdell, A. P., & Cook, R. G. (2005). Two-item same-different concept learning in pigeons. *Animal Learning & Behavior, 33*(1), 67–77.

Blender Online Community. (2017). *Blender—a 3D modelling and rendering package* [Computer software manual]. Amsterdam: Blender Institute. Available from http://www.blender.org.

Bohn, T., Hu, Y., & Ling, C. X. (2019). Few-shot abstract visual reasoning with spectral features. *arXiv preprint arXiv:1910.01833*.

Bond, A. B., Kamil, A. C., & Balda, R. P. et al. (2004). Pinyon jays use transitive inference to predict social dominance. *Nature, 430*(7001), 778–781.

Bongard, M. M. (1970). *Pattern recognition*. New York, Washington: Spartan Books.

Bromley, J., Jamesh, W. B., Léon, B., Isabelle, G., Yann, L., Cliff, M., Eduard, S., ... Roopak, S. (1993). Signature verification using a "Siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence, 7*(4), 669–688.

Byosiere, S.-E., Feng, L. C., Chouinard, P. A., Howell, T. J., & Bennett, P. C. (2017). Relational concept learning in domestic dogs: Performance on a two-choice size discrimination task generalises to novel stimuli. *Behavioural Processes, 145*, 93–101.

Cammarata, N., Carter, S., Goh, G., Olah, C., Petrov, M., & Schubert, L. (2020). Thread: Circuits. *Distill,* San Francisco, Calif. https://distill.pub/2020/circuits.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., ... Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255), doi:10.1109/CVPR.2009.5206848.

Depeweg, S., Rothkopf, C. A., & Jäkel, F. (2018). Solving Bongard problems with a visual language and pragmatic reasoning. *arXiv preprint arXiv:1804.04452*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., ... Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2625–2634).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., & Unterthiner, T. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations, ICLR 2021, Vienna*, https://openreview.net/forum?id=YicbFdNTTy.

Fang, H., Gupta, S., Iandola, F., Srivastava, R. K., Deng, L., Dollár, P., ... Gao, J. et al. (2015). From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1473–1482).

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, 108*(43), 17621–17625.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139.

Funke, C. M., Borowski, J., Stosio, K., Brendel, W., Wallis, T. S., & Bethge, M. (2019). The notorious difficulty of comparing human and machine perception. In *2019 Conference on Cognitive Computational Neuroscience* (pp. 2019–2195).

Giurfa, M., Zhang, S., Jenett, A., Menzel, R., & Srinivasan, M. V. (2001). The concepts of sameness and difference in an insect. *Nature, 410*(6831), 930.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A, ... Olah, C. (2021). Multimodal neurons in artificial neural networks. *Distill, 6*(3), https://distill.pub/2021/multimodal-neurons, doi:10.23915/distill.00030.

Grosenick, L., Clement, T. S., & Fernald, R. D. (2007). Fish can infer social rank by observation alone. *Nature, 445*(7126), 429–432.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1026–1034). Santiago, Chile.

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

Herrnstein, R. J., & Loveland, D. H. (1964). Complex visual concept in the pigeon. *Science, 146*(3643), 549–551.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hofstadter, D. R. (1979). *Gödel, escher, bach: An eternal golden braid.* New York, NY: Basic Books.

Hogue, M.-E., Beaugrand, J. P., & Laguë, P. C. (1996). Coherent use of information by hens observing their former dominant defeating or being defeated by a stranger. *Behavioural Processes, 38*(3), 241–252.

Hoshen, D., & Werman, M. (2017). IQ of neural networks. *arXiv preprint arXiv:1710.01692*.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). Perceiver: General perception with iterative attention. *Proceedings of the 38th International Conference on Machine Learning, PMLR 139* (pp. 4651–4664).

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., & Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2901–2910).

Karpathy, A. (2014). What I learned from competing against a ConvNet on ImageNet, http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/. Accessed July 12, 2021.

Katz, J. S., & Wright, A. A. (2006). Same/different abstract-concept learning by pigeons. *Journal of Experimental Psychology: Animal Behavior Processes, 32*(1), 80.

Kim, E., Hannan, D., & Kenyon, G. (2018). Deep sparse coding for invariant multimodal Halle Berry neurons. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

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

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

Kubilius, J., Schrimpf, M., Hong, H., Majaj, N., Rajalingham, R., Issa, E. B., . . . DiCarlo, J. J. (2019). Brain-like object recognition with high-performing shallow recurrent ANNs. *33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada.*

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., . . . Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*(4), 541–551.

Mårtensson, G., Ferreira, D., Granberg, T., Cavallin, L., Oppedal, K., Padovani, A., . . . Westman, E. (2020). The reliability of a deep learning model in clinical out-of-distribution MRI data: A multicohort study. *Medical Image Analysis, 66*, 101714.

Martinho, A., & Kacelnik, A. (2016). Ducklings imprint on the relational concept of same or different. *Science, 353*(6296), 286–288.

Mercado, E., III, Killebrew, D. A., Pack, A. A., Mácha, I. V., & Herman, L. M. (2000). Generalization of same–different classification abilities in bottlenosed dolphins. *Behavioural Processes, 50*(2–3), 79–94.

Messina, N., Amato, G., Carrara, F., Falchi, F., & Gennaro, C. (2019). Testing deep neural networks on the same-different task. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)* (pp. 1–6).

Mondragón, E., Alonso, E., & Kokkola, N. (2017). Associative learning should go deep. *Trends in Cognitive Sciences, 21*(11), 822–825.

Murphy, G. (2004). *The big book of concepts.* Cambridge, MA: MIT Press.

Nie, W., Yu, Z., Mao, L., Patel, A. B., Zhu, Y., & Anandkumar, A. (2020). Bongard-logo: A new benchmark for human-level concept learning and reasoning. *Advances in Neural Information Processing Systems, 33.*

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., & Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: An empirical study. *arXiv preprint arXiv:1802.08760*.

Oden, D. L., Thompson, R. K., & Premack, D. (1990). Infant chimpanzees spontaneously perceive both concrete and abstract same/different relations. *Child Development, 61*(3), 621–631.

Peer, D., Stabinger, S., & Rodríguez-Sánchez, A. J. (2021). Auto-tuning of deep neural networks by conflicting layer removal. *arXiv preprint arXiv:2103.04331*.

Pepperberg, I. M. (1987). Acquisition of the same/different concept by an African grey parrot (*Psittacus erithacus*): Learning with respect to categories of color, shape, and material. *Animal Learning & Behavior, 15*(4), 423–432.

Pham, H., Dai, Z., Xie, Q., Luong, M.-T., & Le, Q. V. (2020). Meta pseudo labels. *arXiv preprint arXiv:2003.10580*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., . . . Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *2015 International Conference on Learning Representations, Puerto Rico*.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., & Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In *2017 International Conference on Machine Learning* (pp. 2847–2854).

Raven, J. C., & Court, J. H. (1938). *Raven's Progressive Matrices*. Los Angeles, CA: Western Psychological Services.

Ricci, M., Kim, J., & Serre, T. (2018a). Not-so-clevr: Visual relations strain feedforward neural networks. *arXiv preprint arXiv:1802.03390*.

Ricci, M., Kim, J., & Serre, T. (2018b). Same-different problems strain convolutional neural networks. Retrieved May 28, 2018, from http://arxiv. org/abs/1802.03390.

Rozell, C., Johnson, D., Baraniuk, R., & Olshausen, B. (2007). Locally competitive algorithms for sparse approximation. In *2007 IEEE International Conference on Image Processing* (Vol. 4, pp. IV–169).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., . . . Lillicrap, T. (2017). A simple neural network module for relational reasoning. *31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA*.

Schaffer, J. (2015). What not to multiply without necessity. *Australasian Journal of Philosophy, 93*(4), 644–664.

Schrier, A. M., & Brady, P. M. (1987). Categorization of natural stimuli by monkeys (*Macaca mulatta*): Effects of stimulus set size and modification of exemplars. *Journal of Experimental Psychology: Animal Behavior Processes, 13*(2), 136.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations (ICLR'15)*. San Diego, USA.

Stabinger, S., Peer, D., & Rodrguez-Snchez, A. (2021). Arguments for the unsuitability of convolutional neural networks for non–local tasks. *Neural Networks, 142*, 171–179.

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

Stabinger, S., Rodríguez-Sánchez, A., & Piater, J. (2016b). Learning abstract classes using deep learning. In *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly Bionetics)* (pp. 524–528).

Stabinger, S., & Rodríguez-Sánchez, A. J. (2017). Evaluation of deep learning on an abstract image classification dataset. In *ICCV Workshops* (pp. 2767–2772).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the 2015 IEEE conference on Computer Vision and Pattern Recognition* (pp. 1–9).

Teney, D., Anderson, P., He, X., & van den Hengel, A. (2018). Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4223–4232).

Teney, D., Wang, P., Cao, J., Liu, L., Shen, C., & Hengel, A. V. D. (2019). V-PROM: A benchmark for visual reasoning using visual progressive matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(7), 12071–12078.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA*.

Vogels, R. (1999). Categorization of complex visual images by rhesus monkeys. Part 1: Behavioural study. *European Journal of Neuroscience, 11*(4), 1223–1238.

Vonk, J., & MacDonald, S. E. (2002). Natural concepts in a juvenile gorilla (*Gorilla gorilla gorilla*) at three levels of abstraction. *Journal of the Experimental Analysis of Behavior, 78*(3), 315–332.

Vonk, J., & MacDonald, S. E. (2004). Levels of abstraction in orangutan (*Pongo abelii*) categorization. *Journal of Comparative Psychology, 118*(1), 3.

Wang, K., & Su, Z. (2015). Automatic generation of Raven's Progressive Matrices. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*.

Wasserman, E., DeVolder, C., & Coppage, D. (1992). Non-similarity-based conceptualization in pigeons via secondary or mediated generalization. *Psychological Science, 3*(6), 374–379.

Wasserman, E. A., Castro, L., & Freeman, J. H. (2012). Same–different categorization in rats. *Learning & Memory, 19*(4), 142–145.

Wright, A. A., & Katz, J. S. (2006). Mechanisms of same/different concept learning in primates and avians. *Behavioural Processes, 72*(3), 234–254.

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., & van den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding, 163*, 21–40.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., . . . Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the 32nd International Conference on Machine Learning, PMLR 37* (pp 2048–2057).

Yun, X., Bohn, T., & Ling, C. (2020). A deeper look at Bongard problems. In *Proceedings of 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020 Ottawa. LNAI 12109* (pp. 528–539).

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zentall, T., & Hogan, D. (1976). Pigeons can learn identity or difference, or both. *Science (New York, N.Y.), 191*, 408–409.

Zentall, T. R., Galizio, M., & Critchfield, T. S. (2002). Categorization, concept learning, and behavior analysis: An introduction. *Journal of the Experimental Analysis of Behavior, 78*(3), 237–248.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *5th International Conference on Learning Representations, 2017, Toulon, France*.