CYCLICAL BOTTOM-UP TOP-DOWN NEURAL NETWORKS FOR RELATIONAL REASONING

A PREPRINT

Aviv Netanyahu
netanyahu.aviv@gmail.com

Shimon Ullman
shimon.ullman@weizmann.ac.il

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel

ABSTRACT

Reasoning about relations between objects in images is key to scene understanding. Today, this task is still difficult for Artificial Intelligence to master. Detecting the objects in a scene is not enough for building an understanding of complex situations such as '*stealing*' (Fig. 1). However, object recognition combined with semantic information about the *interactions* among them has a better chance of unraveling the puzzle of a scene.

True reasoning comes with the ability to generalize broadly to different tasks and distributions, since the objective is solving a defined problem and not learning a dataset. For instance, in order to understand the relation 'right of', it should be possible to establish the relation between objects never encountered during training. If train and test data distributions are similar, an expressive model with enough data could succeed on the test set. It is important therefore to test on *novel distributions* or *novel tasks*. Moreover, it is difficult to obtain a sufficient amount of training data for the extremely large set of possible objects in each relation, which also increases over time.

In this work we focus on several aspects of *relational reasoning*. In particular, given some object A and a relation, the goal is to identify an object B related to object A, in a given scene, in the specified relation, while being able to generalize to other objects. Additionally, it is important to represent relations meaningfully, such that visually similar relations have similar representations. This is learned and tested on synthetic data such as handwritten characters and shapes.

The method we propose using for dealing with relations is a *bottom-up top-down* deep neural network. Using the proposed architecture allows us to introduce task *selectivity*, as opposed to extracting all objects and relations from an image, regardless of the task. Moreover, we show that this architecture allows better generalization than simple feedforward models.

The network can further be used in a *cyclic fashion* for answering visual questions, where the output of each cycle is based on an object related to the object from the previous cycle. Other uses for applying the model in a cyclic fashion can be scene graph and image caption generation, in which case a policy for choosing the initial object and a relation in each cycle is needed.

***K*eywords** relational reasoning · generalization · bottom-up top-down deep neural network

1 Introduction

Scene perception is one of the fundamental tasks of artificial visual intelligence. Research in this area takes form in pursuing several related tasks including image captioning, scene graph generation and visual question answering (VQA). In order to interpret what goes on in an image, the system needs to have knowledge about the objects present in the scene and their properties and interactions, as well physical knowledge about the world. Whereas single object recognition is almost a solved task [1], understanding relations between objects is one of the main next steps towards significant progress in machine intelligence [2]. In addition to being an important part of general artificial visual intelligence, detecting and reasoning about object relationships is a required aspects in various applications, such as helping the blind and visually-impaired in day to day conduct and integrating VQA into image retrieval systems [3].

In order to verify that a model can reason about relations, it is critical to test it on a **generalization set**. Consider the man grabbing a bag in Fig. 1. Without ever seeing this specific bag, it is still possible to identify the 'grabbing' relation and the person involved in this relation, and understand the possible consequences. The ability of models to detect 'grabbing a bag' is often tested using a training and test sets using instances of grabbing-bag images. A more meaningful test, however, should involve different types of objects altogether, i.e., grabbing objects that are not bags. This generalization capacity is important since the task is to understand the relation, ('grabbing', in this case), which is derived essentially from spatial relations between the objects and from image features generated by the interaction between objects, and not from the objects themselves. Humans have such a generalization capacity and can identify 'grabbing' of novel objects, and intelligent vision systems should have a similar capacity.

Figure 1: 'Stealing'. A typical human description of the scene is: A lady sitting on a chair with a bag hanging over the back of the chair. A man is approaching from behind the chair, grabbing the bag.

When **humans observe a scene**, they do not grasp all details simultaneously, but build an understanding of the situation step by step. Ongoing experiments in the lab demonstrate that when presenting humans with images for short periods of time, they only pick up on partial information, which then develops systematically over time. What allows humans to establish the understanding is the constant feedback from our cognition to the visual system, based on what we see. Consider Fig. 1. People tend to first perceive humans in a scene, i.e., the man or the woman in this case. Say we start focusing on the man. We can easily observe, by his pose, that he is grabbing something, thus we focus on the bag. Since the bag is hanging over the back of the chair, we come to notice a lady sitting on that chair. We infer that the bag is hers, and we might notice that her gaze is not fixated on the man, hence he is probably stealing the woman's bag. Since it is infeasible to extract all the possible relations in a scene with many objects, we want a model that extracts relations in a guided manner similarly to the human perception. This means that scene interpretation is an extended process, which allows the extraction of relevant relations between selected objects. This behavior can help answer relational questions, e.g., by guiding attention from one object in the scene to another based on the relations between these objects. More importantly, this could serve as a stepping stone towards the understanding of complex image scenarios, such as 'stealing'.

Feedback **in the brain** from our cognition to the visual system takes place via bottom-up (BU) and top-down (TD) connections in the visual cortex. Both BU and TD feedback occur between different levels in the visual cortex and at each level within its layers [4]. As soon as a human observes the environment, sensory input is continuously processed by the visual cortex according to the BU principle; visual information that enters through the eyes, flows from lower to higher visual areas, i.e. from bottom to top. But how do we know whether or not a given piece of information is more important than another? The TD principle helps us determine this, based on information that flows in the opposite direction, i.e., from top to bottom; to direct attention to particular stimuli, for example. The brain uses previous experiences to organize information in the present context and to make predictions on this basis. In other words, the TD flow influences the BU flow and steers our *attention* towards objects that are important in the current situation [5]. This can happen automatically, for example due to the sudden appearance of a threatening stimulus. Or it can also happen through attention, for example when we are looking for something based on our *prior knowledge* of the world, *primed* by a previous stimuli for instance the plot of a movie, or *following instructions* [6].

2 Related work

We survey related other work that deals with relational reasoning. We start with *implicit end-to-end* approaches, i.e. which analyze scenes without explicitly modeling the processes of extracting relations using black-box architectures. We then discuss *non-selective* (or *comprehensive*) methods that examine all objects and relations, and *selective* methods

that integrate attention. Finally, we review models similar to ours, as far as the use of the *BU-TD architecture* is concerned.

It is possible in some cases to achieve good results for tasks requiring relational reasoning without directly acknowledging the relations between objects of interest. This can be done by using **implicit end-to-end** approaches. Most approaches for VQA, generating a natural language answer to a natural language question about an image, combine CNNs and LSTMs in various ways, trained end-to-end with large data sets of questions and answers [7, 8, 9]. Image captioning, i.e., generating automatically a textual description of an image, can also be done end-to-end, without acknowledging specific related objects, by training on a dataset of (image, caption) pairs. Such methods combine CNNs and LSTMs like in [10, 11]. When using language priors, generalizing is more challenging. We propose a different approach, with an emphasis on learning relation representations from visual data. Other end-to-end methods for image captioning attempt to map inputs to outputs using attention [12, 13], implicitly modeling the reasoning processes (via the chosen architecture), or using reinforcement learning [14, 15] without explicitly modeling the reasoning processes at all. Scene graph generation, the task of generating a graph that represents an image, generates a graph with nodes that correspond to object bounding boxes and their object categories, and its edges correspond to pairwise relationships between objects. Work like [16] uses a CNN and RNNs to predict the scene graph using an end-to-end approach as well. The RNNs are used iteratively to improve the scene graph prediction at each iteration.

The **end-to-end** models are often **non-selective**, in the sense that they try to extract all the objects and all possible relations in the scene. Work such as [17] for relational reasoning has shown super-human performance on a limited subset of questions applied to the CLEVR dataset [18]. However, it requires implicitly analyzing all object pairs in an image and their corresponding relations, which results in a computational burden that may not be required for many tasks. Also, this approach does not identify explicitly the objects and their relations, and it is unlikely to generalize to more complex tasks and data. There are also **non-selective** models, that are **not end-to-end** such as [19] which extracts object proposals and then applies relation detection to all pairs using language priors. Work on recognizing human-object interactions [20] uses a relation attention mechanism; however, it computes this for each human and possible relation, and compares the outcome with each object. We prefer pursuing a selective approach.

Selective methods do not extract all relations in the scene, but can focus on relations between selected objects only. More recent work on referring relationships [21] suggests that the visual appearances of relations are too varied to learn, and focuses on localizing the two objects in a given relationship, conditioned on one another. It is selective in the sense that it attends to the object in the specified relation out of multiple entities of that same object. However, it does not distinguish between multiple entities of the same object in the same relation. Most successful methods for VQA use also an *attention* mechanism [22, 23]. This should have a similar effect to inputting features of interest as we propose (see Fig. 2). Other end-to-end methods that are selective by the models' intermediate outputs have also shown good results on CLEVR [24, 25]. However, these methods use a compositional approach which enables *combinatorial generalization* [26], but might not scale well to a dataset with many relations.

There are other artificial models that use a **BU-TD architecture**, for instance fully convolutional networks (FCN) [27] and Mask R-CNN [28] for semantic segmentation, and the U-Net convolutional networks for biomedical image segmentation [29]. The uniqueness of our BU-TD proposed architecture is the additional TD input which enforces attention. Another major difference is that FCN, Mask R-CNN and U-Net use lateral connections only from BU to TD, whereas we use lateral connections in the opposite direction as well. Moreover, the choice of architecture details, such as which layers form the lateral connections, how they are connected with the TD layers and if their weights are fixed or learned. Also, the choice of using fully connected layers versus pooling, etc.

3 The Counter Stream model for relational reasoning

3.1 Overview

The proposed scheme is an artificial BU-TD architecture inspired by the BU-TD cortical structures. This architecture allows information from high-level areas to control and guide the extraction of information from lower-level stages. In the current work, the combination of BU and TD processes is used to deal with relationships between objects. For example, given an object and a relation, find the second object that satisfies the given relation. We will also show examples where the BU-TD scheme helps to generalize the classification of relations from trained to novel objects, as well as examples where the relations themselves can be generalized to form novel relations.

Within the BU-TD scheme, the feedforward BU part is a convolutional neural network (CNN) which receives an image as input. The BU part is followed by a feedback TD architecture, which is connected to the BU part via lateral connections (see Fig. 2). The TD architecture receives additional so-called 'guidance' input. For example, in extracting a relation, the TD instruction will specify a subject and a relation of interest, and the output of the next BU pass

will then output the object in the image which participates in the specified relation with the subject. We will show examples where our network can be used in a cyclic fashion, extracting more than a single relation, where a cycle is formed of a BU pass followed by a TD pass. The number of completed cycles is equal to the number of relations in question, where the network is expected to output the next related object at the end of each cycle. The input subject to each cycle is the previous cycle's output object. In practice, the network we train and test usually consists of a cycle and a half; the first BU pass followed by a TD pass form one cycle, and the additional BU pass, implemented by additional layers to the architecture with residual connections to the TD part, forms an additional half a cycle. We refer to the above cycle-and-a-half structure as the *counter stream* (CS) model in Fig. 2. This is an unfolding in time, of a network composed of the interconnected BU and TD streams. This forms a recurrent network, in which the computation continues to cycle between the BU and TD networks. The weights of BU1 and BU2 are therefore identical. Training is performed by back-propagation though the unfolded network. This allows the scheme to perform a first BU pass, analyze the results obtained from the image, select what to extract from the image next, and guide the extraction of the selected information in the subsequent BU pass.

In terms of images, We use initially EMNIST [30] digits, and then show that our method generalizes to EMNIST handwritten English letters, as well as handwritten letters in various languages (Omniglots [31]), and other general shapes. In terms of relations, we focus initially on spatial relations, i.e. 'right', 'left', 'above' and 'below', and generalize to all eight principal directions of the compass rose. As we shall see, the factors that support broad generalization in our relation processing come from the architecture of the CS, especially its lateral connections, running it in a cyclic fashion over a sequence of relations, and separating object classification from spatial relational reasoning.

Our approach is different from other approaches to relational reasoning, in terms of the following three main issues:

1. The first issue concerns *implicit* versus *explicit* representation of objects and relations. This work uses explicit representations as input to the CS model.

2. The second issue has to do with detecting *all* objects and relations versus *selecting* only those of interest for performing a scene perception related task. When there are many objects in an image, extracting all objects and relations is redundant for the purpose of answering (most) relational questions or identifying the main event in an image. The CS model is selective, guiding the process to selected objects and relations. subject and relation guidance.

3. The third issue concerns the computational approach, i.e., *end-to-end* versus *sequential* detection of objects and relations. This work takes the sequential approach, which breaks problems requiring relational reasoning into smaller units composed of two objects and the relation between them. This grants model interpretability and better generalization.

3.2 The Counter Stream (CS) model

Given an image, an object A in that image and a relation, our initial goal is to output an object B in the image, which is related to object A via the specified relation. For this task we use the CS model, which is illustrated schematically in Fig. 2 and consists of three parts: BU1, TD and BU2.

The BU1 part is a standard CNN structure, and it serves to extract meaningful features from the image which can be used for preliminary classification or localization of objects in the image. The images are inputted into the BU1 pillar composed of convolutional layers and a fully connected layer. It is possible to add a second fully convolutional layer (illustrated below) as an intermediate output, whose loss is added to the BU2 loss. This intermediate output is the multi-label classification of all objects in the image or their locations. Instead of convolutional layers, another version of the BU1 is made up of resnet blocks [32], specifically an implementation of renet-18 was used.

The TD's purpose is guiding the next BU stage, for fulfilling a specified task. The TD pillar receives an instruction 'flag' as input, with an encoding of an object A and a relation. These encodings can be one-hot encodings or more meaningful representations such as object A's features. This flag is embedded by a fully connected layer, which is concatenated with the first fully connected BU1 layer, and embedded again by a fully connected layer. The next TD layers are convolution or *transpose convolution* (also known as *deconvolution*) layers, each connected with an equivalent side layer. The equivalent side layers are products between matching layers in shape and hierarchy from the BU1, and learnt weights (per node or channel). TD layers are convolutional if there is no increase in dimensionality; if there is, transpose convolution layers are used to learn this upsampling, as used in FCNs for Semantic Segmentation. The last TD layer is equivalent to the first BU1 layer (which is the result of convolving the input image once). It is optional to add another layer, upsampling to the original image size, for segmenting object A given by the flag, or an object B related to it by the given relation. The segmentation loss is added to the BU2 loss (and the BU1 loss if it exists).

The BU2's purpose, similar to BU1, is extracting meaningful features from the image, however for classifying or locating a *specific* object B as opposed to general objects. The BU2 pillar shares its convolution filters and structure with the BU1. Its first layer (which is not trainable) is identical to the BU1's first layer. The fully connected weights are not shared, neither is batch normalization. Side layers from the TD are connected to the BU2 in the same fashion the BU1 side layers are connected to the TD. The BU2 output is an object B's class or location, which is related to object A by the relation given to the TD.

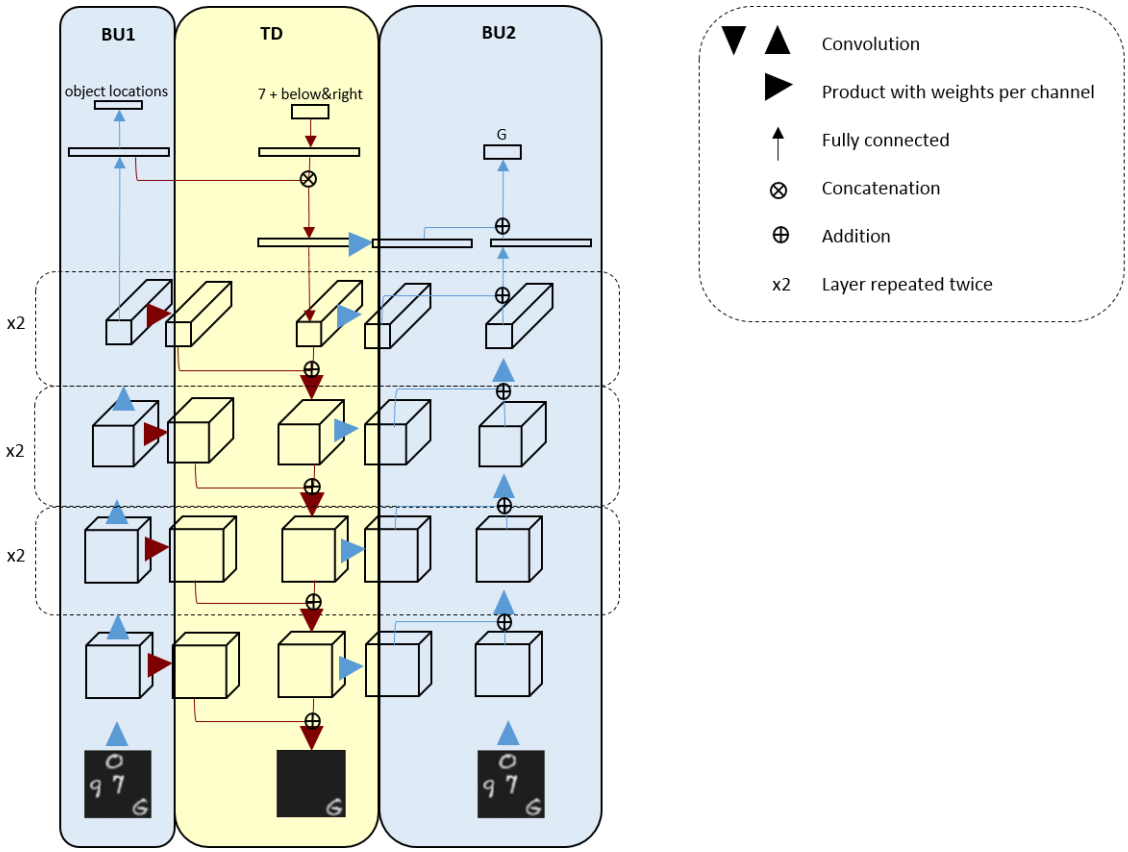

Figure 2: The CS model consists of a BU1 pillar, which receives an image as input, a TD pillar, which receives an instruction flag as input with an object A and relation (e.g. the location of 7 in the image and the relation '*below&right*'), and a BU2 pillar, that outputs the related object B accordingly (e.g., the location of *G* in the image). The CS model may yield two additional outputs: (1) An additional BU1 output of the locations of all objects in the image, and (2) an additional TD output of segmentation of the its flag object A (i.e., 7) or of its related object B (i.e., *G*).

4 Experiments

In this section we describe several experiments that deal with the extraction of an object given another object and their relation. All experiments, but the last, use images with EMNIST characters, Omniglots and shapes with spatial relations. The last experiment uses natural images and human-object interactions, in particular the 'riding' relation. We use EMNIST since it is a simple but still challenging domain that was used extensively in past development of Deep Neural Networks. Furthermore, it allows us to generate large and well-controlled datasets and label them. In the next subsection, 4.1, we describe the datasets we used. Then, in subsections 4.2 - 4.6, each experiment will be described and discussed.

4.1 Datasets

4.1.1 Concatenated characters

Each image is a horizontal concatenation of four different characters of size 28×28. The final image sizes are 28×112 (see Fig. 3). This dataset has several variants, where characters are handwritten digits between $0-9$ and English letters from the EMNIST Balanced Dataset, five Omniglot languages or shapes.

Figure 3: An image from the concatenated characters dataset, which consists of four characters in a row.

4.1.2 Scattered characters

Images are of size 64×64, and contain two to four different characters of size 21×21 (see Fig. 4). Like the dataset described above, this dataset has several variants, where characters are handwritten digits between $0-9$ and ten capital English letters from the EMNIST Balanced Dataset, five Omniglot languages or shapes. The location of the *first character* (i.e. its center coordinates) is randomly selected. The *second character* location is also randomly selected under the constraint that there is a certain spatial relation between it and the first character. Object B (the second, or 'target' character) is 'right of' object A (the first, or 'reference' character) in the sense that A and B's center rows are within distance of at most 5 pixels, B's center column is right of A's, and that the 21×21 characters windows do not overlap. The placement of B is defined in a similar manner for 'left', 'above' and 'below' relations as well. The definition for the four possible diagonal relations (such as 'above & right') is:

$$|row_{char1} - row_{char2}| \quad = \quad |col_{char1} - col_{char2}|$$

in each respective direction. The choice for the *other* zero to two character locations is random under the constraint that there is no overlapping with previously placed characters, and no interfering with the relation between the first and second characters. These characters may or may not be related to previously placed characters in the image in any of the eight interactions described.

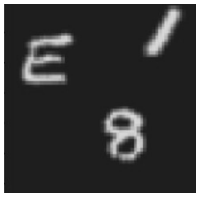

Figure 4: An image from the scattered characters dataset, which consists of two to four EMNIST characters in a scatter.

4.1.3 HICO-DET subset

We used a subset of the Humans Interacting with Common Objects Detection (HICO-DET) dataset [33]. This subset includes RGB images where the interacting object is visible and the interacting human and object's width or height are at least 50 pixels. Interacting objects are $\in$ {bicycle, motorcycle, elephant, horse, cow, giraffe, sheep, skateboard, skis, snowboard, surfboard}. The interaction relations include 'riding' and other 'non-riding' interactions such as 'repair', 'wash' and 'carry'. We split 'riding' into four relation types based on visual similarity of humans in these interactions: type 1 - riding bicycles and motorcycles, type 2 - riding elephants, horses, cows, giraffes and sheep, type 3 - riding skateboards, skis, snowboards and surfboards, type 4 - 'non-riding'. We split the objects in each 'riding' type to train and generalization test sets, and increased the size of type 2 generalization set to a reasonable size by adding another 35 images of humans riding cows, giraffes and sheep using Google Image Search. These images were used for testing generalization of 'riding' to objects that were not seen during training. Additionally, we increased the size of the train set by moving 200 'non-riding' images from the test set to the train set.

4.2 Object and spatial relation generalization on synthetic data

The main objective of the following experiment was to extract an object given another object and their relation, and testing generalization to novel objects and relations. In our first generalization test, we tried to exploit the similarity of

train and generalization objects for generalizing. Specifically, we used a CS network on the concatenated characters dataset with 'right' and 'left' relations. We trained on digit images to extract target object B's class, and tested generalization to images with novel object A (reference) letters. This was possible since we used object A's *features* as TD flags, and since visually similar digits and letters (e.g. 8 used in train and *B* in generalization test) have similar features. A limitation of this method is that it cannot generalize to novel target object Bs, since we extract B's *class*, which was not used in training. Moreover, it will not be able to scale well to substantially different reference object As from the ones used during training. Therefore, in the following experiment we describe, we separated between the tasks of object classification and relational reasoning. For the relational reasoning part, we used the object *locations*, independent of their identity. We show generalization to object types (i.e., from digits to characters) and to additional spatial relations (from right, left, above and below relations, to diagonal directions).

Given a reference object A's class and a relation, the goal was to output the related object B's class. For humans, such generalization is natural; we can tell that B is to the right of A, even if one of them, or both, are novel objects that were not examined in left-right relations in the past. In order to accomplish this *and* generalize to new objects, we divided the task into **three tasks**: 1) translating object A from its class into its location, 2) outputting object B's location given object A's location and their relation, 3) translating B's location to its class. In this way, the classifiers (tasks 1 and 3) are separated from the relational reasoning (task 2). Tasks 1 and 3 are trained on all classes of interest, but task 2 is trained only on a subset of these objects. We can generalize to novel objects A and B which may be objects that the system was trained to classify, but never participated in the relational training, or, they can be entirely novel, i.e. never seen by the system before. If the classes are known, we can test whether the system can still perform the task. If the objects are entirely novel, then it is still possible to perform task 2, but it will need A's position in the input, and will produce B's location rather than its class. In order to classify object B, after extracting its location, task 3 would need further training. To avoid *catastrophic forgetting* (abruptly loosing knowledge of previously learned tasks as information relevant to the current task is incorporated), the three tasks were trained simultaneously on the same network. Another option, would be to train on an unfolded three task CS network, i.e. BU-((TD-BU)$\times$3) and train on samples that consist of the three tasks. In this way, the TD parts in tasks 2 and 3 will receive lateral input from tasks 1 and 2 BU2 parts respectively, whereas currently, they receive BU1 lateral input. We leave this for future exploration.

In the following series of experiments, we implemented this, training the CS model to learn spatial relations: 'right', 'left', 'above' and 'below', on the scattered characters dataset and generalizing to new objects and diagonal relations. During **training** (see Fig. 5 (a)), the BU1 input to task 1 is an image, and the TD input is the class of the reference object A. The BU2 output is object A's location. For task 2, the BU1 input is the image, the TD input is a relation and object A's ground truth location. The BU2 output is object B's location. In task 3, the BU1 input is the image, the TD input is object B's ground truth location, and the BU2 output is object B's class. In all tasks, the BU1 initial objective is the location of all characters in the image (i.e. two to four pairs of (x,y) coordinates). For consistency, the coordinate order of objects is from left to right, top to bottom in the image. The TD objective in all tasks is segmentation of the object given as the TD flag (as a class or location). The image inputs in tasks 1 and 3 are of digits and letters, and in task 2 only digits, in order to test generalization to letters. **Testing** the can be done on each task separately using ground truth locations in tasks 2 and 3 TD flags. Another option is performing the three tasks *sequentially*, where the BU2 output of each task provides the TD input to the next. In this case, the input is the image, object A's class and a spatial relation, and the expected output is object B's class (see Fig. 5 (b)).

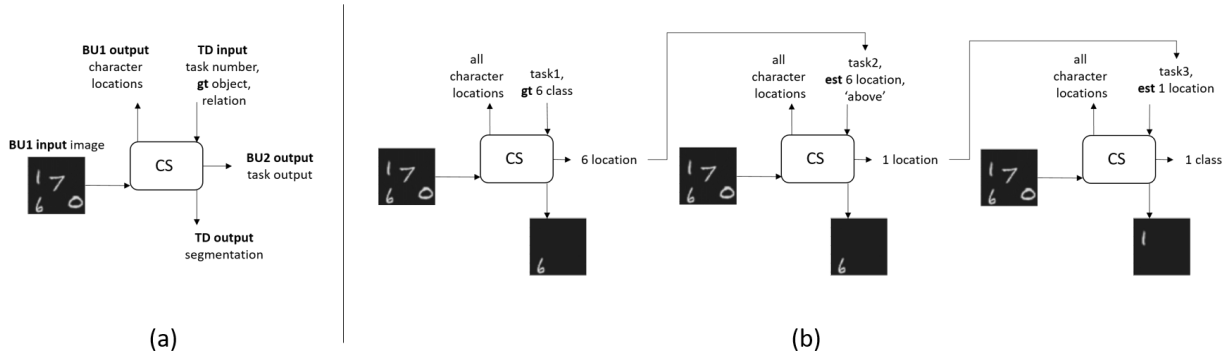

Figure 5: Summary of the three tasks. (a) Training the tasks simultaneously. (b) Sequential testing example.

Objects are represented as one-hot vectors of size 20: 10 digits and 10 letters. **Locations** are concatenations of two one-hot vectors which represent the (x,y) coordinates of the character centers. There are 44 options for each coordinate

and 44^2 location options. **Relations** are represented as directions in $\mathbb{R}^2$: 'right' as $[1,0]$, 'left' as $[-1,0]$, 'above' as $[0,1]$ and 'below' as $[0,-1]$. Diagonal relations are sums of two directions, i.e. 'above & right' is $[1,1]$, and so on for 'above & left', 'below & left', 'below & right'.

The **loss** of tasks 1 and 2 (which output locations) is the sum of softmax cross-entropy on each coordinate separately. The way the scattered characters dataset is created, promises there is only one correct answer, i.e. there is only one 'right' object, which ensures task 2 is a classification problem and not a multi-label problem. Task 3's loss is softmax cross entropy on the outputted class. The BU1 initial loss is the sum of softmax cross-entropy on all coordinates. The TD intermediate loss is the mean squared error (MSE) loss.

The **results** on variants of the scattered characters dataset (the test, and object and relation generalization sets) are shown in Tab. 1. Accuracy for tasks 1 and 2 is calculated for correct classification of rows *and* columns. Allowing a margin for mistake of even 1 pixel increases accuracy by a few percentages for all datasets. For task 3, accuracy is on object classification. For each dataset, each task is evaluated separately with correct inputs, and as a sequence. A sequence means the estimated output of each task provides the input to the next task (and not the ground truth). The **test** set contains 'right', 'left', 'above' and 'below' relations, digit and letter images for tasks 1 and 3 and only digit images for task 2. Each task is evaluated separately on this dataset. The **object generalization** set is similar but with digit and letter images for task 2. The **relation generalization** set contains only diagonal relations: 'above & right', 'above & left', 'below & left' and 'below & right' and only digit images. The **object** & **relation generalization** set contains only diagonal relations and digit & letter images. The generalization datasets are evaluated separately and as sequences. On task 2, there is almost perfect generalization to objects. On diagonal relations, task 2 generalization is not as good, however, this result shows that the chosen relation representation is meaningful. Note that multiplying the accuracies of each task separately is not equal to sequential task 3 accuracy. However, allowing a mistake margin of several pixels for each task, and then multiplying accuracies is roughly equal to sequential task 3 accuracy on each dataset.

Table 1: Object & relation generalization BU2 accuracy.

Dataset	task 1	task 2	task 2 sequence	task 3	task 3 sequence
random	0.0005	0.0005	-	0.05	-
test	0.91	0.98	-	0.93	-
object generalization	0.91	0.97	0.92	0.93	**0.88**
relation generalization	0.91	0.82	0.78	0.94	**0.78**
object & relation generalization	0.9	0.79	0.74	0.93	**0.77**

Since all tasks were learned simultaneously, tasks 1 and 3 on digits and letters, and task 2 only on digits, we checked if tasks 1 and 3 affect task 2 and help improve its generalization performance on letters. To do this, we created a modified version of the object generalization dataset where objects A and B in all images are replaced by Omniglots or general shapes (see Fig. 6). Task 2 accuracy is only 18%, however for a mistake margin of 5 pixels per row and/or column, accuracy is 94% which is close to task 2 accuracy on the object generalization dataset (97%). This can be partially explained by the average center of the Omniglots and shapes, which is shifted by 2 pixels from the average EMNIST digits and letters center in the 21×21 character window. Both object A and B are replaced, potentially causing a 4 pixel shift on average. Small mistakes in location estimation (2-3 pixels for row and column estimations) are only slightly significant for task 3 accuracy where B is a trained class, and cause less than 10% deterioration. This, and the examples in Fig. 6 demonstrate that the *spatial relations are learned regardless of the training objects, but do depend on their general statistics*, i.e. scale, connectivity etc.

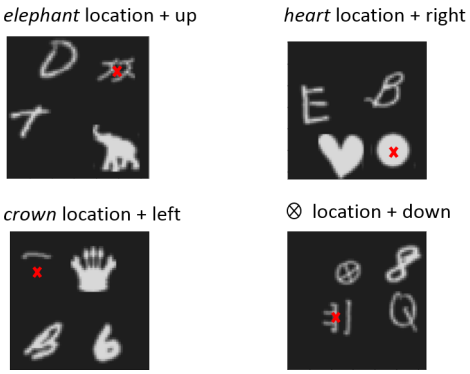

Figure 6: Task 2 object generalization to Omniglots and shapes. The estimated object centers are marked in red.

In this subsection we presented results for object generalization and relation generalization. For object generalization, the CS scheme uses a sequence of instructions to compute spatial relations between locations, rather than specific objects. As a result, the scheme can generalize to completely new objects. For relation generalization, the scheme generalizes at least in part from the trained relations to novel relations, which are intermediate between two trained relations.

4.3 Combinatorial generalization

Combinatorial generalization is the ability to generalize in test to (input, output) combinations that were not seen together during training, but did appear in different combinations. We discuss combinatorial generalization in the context of the three tasks described in the previous subsection. Then, we compare combinatorial generalization in the CS model to a BU under similar conditions.

1. Despite tasks 1 and 3 (described in subsection 4.2) imperfect performance, there is combinatorial generalization to new (character, location) combinations: about 60% of the task 1 test set are combinations that do not appear in the task 1 train set; the same for task 3. For task 2, 90% of the test set (reference flag location, target location) combinations are not in task 2 train set.
2. Next, we describe a simpler experiment we conducted, which shows that the CS performance is better than the BU in terms of combinatorial generalization. The BU model we refer to here is a simple feedforward architecture without an additional TD input.

   In order to conduct a fair comparison between the CS and the BU models, the following must be taken into account. First, the output must be of the same nature (e.g. both models should output a related object's class rather than one outputting a class and the other an object segmentation). However, since the BU has no specific instruction with a specific relation, it must output all related objects of all the characters in an image. Moreover, the number of parameters, epochs and training data in each model must be comparable.

   Therefore, we compared the CS to a BU with the same number of layers (the BU is like BU1 in Fig. 2, where each group has 6 convolution blocks instead of 2), trained for the same number of epochs (35) and on the same number of training samples ($50,000$) from the concatenated characters dataset. It is also possible to conduct the following experiment on a CS model without TD input, however we chose a simple BU model, and demonstrate the effect of the CS lateral connections in subsection 4.5. The number of parameters of a CS without TD input and an expanded BU are still on the same scale (57M and 27M respectively), but due to removing the lateral connections, these models are not equal in their complexity. Moreover, it is debatable whether the number of training samples should be equal, since each BU sample essentially contains 8 CS samples ('right of' and 'left of' for each of the four digits). We use the same number of samples, but using 8 CS samples for each BU sample is also an option.

   We compared learning in the BU vs. the CS network using the concatenated characters dataset with 'right of' and 'left of' relations, and 10 digits and 10 letters as objects. In training, 14 out of the 190 possible pair combinations are omitted (each character is in at most 2 omitted pairs), and test evaluation is on these pairs. Since we do not test generalization to *new* data in this setting, we do not use three tasks going via computing locations, but use a single BU-TD-BU cycle (using the reference class and relation as TD input, and expecting the target class as BU2 output). Instead, the input to the CS is an image, a character class and a relation, and the output is the related object's class. The BU input is an image, and the output is a tensor of size $2 \times 20 \times 21$

where:

$$(BU_{output})_{\mathbf{1}ij} = \begin{cases} 1 & j\,is'\mathbf{right}\,of'\,i \\ 0 & else \end{cases}, \quad (BU_{output})_{\mathbf{2}ij} = \begin{cases} 1 & j\,is'\mathbf{left}\,of'\,i \\ 0 & else \end{cases}$$

The 21^{st} bit indicates there is nothing to the right or left of character i or that character i is not in the image. The loss is a sum of softmax cross-entropy applied to 2×20 predictions. Testing on images with the 14 omitted pairs, Hamming distance is 97%, accuracy over neighbors in images which are not the omitted pairs is 93% and *over omitted pairs* 0%. For the CS, accuracy over all samples is 95% and *over omitted pairs* 92%, i.e. the **CS generalizes to seen data in new combinations, whereas the BU doesn't**.

This might not be surprising since the weights connecting to $(BU_{output})_{1ij}$ and $(BU_{output})_{2ij}$ for each omitted pair (i, j) are not updated during training. It might be optional to output three 'right of' characters and three 'left of' characters instead, in order to create a full understanding of the relations in the image and avoid the 'unused neuron' problem. On the other hand, in the CS model the likelihood in train of a TD flag and output from the omitted pairs is zero, but the model is still successful on these pairs.

In conclusion, the CS generalizes to unseen train combinations in the three task setting, and has an advantage in this sense over simple feedforward models with no TD guidance.

4.4 Multi-cycle use of the CS model

A potential use of the CS model could be for VQA tasks, where an answer is obtained by applying an appropriate sequence of TD instructions. We demonstrate this using a single-task CS as described in 4.3 in a repeating fashion for answering simple questions. For each sample, the question is if for some digit in the image, there is another given digit, at any location to the right of the first. For example, for Fig. 3, possible questions are if there is a 3 to the right of 4, to which the answer is *no*, or if there is a 0 to the right of 2, to which the answer is *yes*. The input to the CS is a digit in the image and a 'right of' TD instruction. If the output is the target digit in the question, the model returns 'yes'. Otherwise, the output becomes the next input to the TD path, and this goes on until the digit is found. If the model returns that there is no digit to the 'right of' request, it i.e. reached the last, rightmost digit in the image, 'no' is returned. We limited this procedure to four iterations' since there were four digits in each image. Out of 4980 questions, the model answered 99.4% correctly, does not give an answer for less than 0.5%, and answers falsely for even less.

4.5 CS model: removing lateral connections

In order to examine the role of lateral connections, we repeated the experiment described in 4.2 on the scattered characters dataset, once without the BU1-TD lateral connections (except the top most connection to maintain the model's connectivity) and again without the TD-BU2 connections (except the bottom most connection). Results are shown in Tab. 2. Without **BU1-TD** lateral connections, object and relation generalization reduces substantially. This demonstrates the importance of these connections, possibly for attention applied at the end of the TD part, directed to the relevant location. On the other hand, without **TD-BU2** lateral connections, in this experiment, the object and relation generalization were not affected. Possibly this is due to the relative simplicity of the task, but requires further investigation. We explored this further by looking into *readouts* from the top of each of the three BU2 groups and the feature layer (i.e., the penultimate BU2 layer).

Table 2: The effect of lateral connections on object and relation generalization. Each model is tested on the three generalization sets sequentially. Accuracies are of BU2 for task 3.

model	object generalization	relation generalization	object & relation generalization
CS	0.88	0.78	0.77
CS without BU1-TD	0.75	0.56	0.54
CS without TD-BU2	0.87	0.81	0.75

Readouts are extracted from four layers of the CS model described in 4.3 on the concatenated characters dataset with digits, and from a similar model trained the same way without TD-BU2 connections. We extracted these layers for 1000 samples where the TD task is 'right of', and checked if it is possible to extract the 'right of' and 'left of' digits using simple classifiers. The rationale is that if a simple classifier can extract the answer, then that information is explicitly represented in these layers. We chose a non-linear SVM as the classifier for the feature layer with a RBF kernel. The intermediate layers are not vectorized in the same way as the feature layer, therefore we learned a model with a 1×1 convolution layer for flattening, ReLU for non-linearity, a fully connected layer and softmax cross-entropy for the loss.

Since the original task is 'right of', an indication of a 'good' classifier is its success on the 'right of' task given enough data. 1000 feature layer readouts do not produce good 'right of' results on the model without TD-BU2 connections. Therefore, we extract 2000 samples for the feature layer in each model. Tab. 3 shows that the ability to perform the 'left of' task in the full CS model improves from group 1 throughout groups 2 and 3 up to the feature layer that achieves 74% accuracy which is well over chance (9%). In the model with no TD-BU2 connections, this ability does not improve as a function of progressing the BU2.

Table 3: Readouts from 'right of' samples classified on the 'right of' task and the 'left of' task. The top part shows the train and test results for the full CS model. The first two rows are accuracy on the 'right of' task, the bottom two rows are on the 'left of' task. For each row we display results for each extracted layer. The bottom part shows results for the model without TD-BU2 lateral connections. Training is on 800 samples, testing is on 200.

model	task	set	top1	top2	top3	features
CS	right of	train	1	1	1	1
		test	0.96	0.99	1	1
	left of	train	0.3	1	1	0.85/0.9*
		test	0.3	0.69	0.66	0.49/0.74*
CS without TD-BU2	right of	train	1	1	1	0.89/0.97*
		test	1	1	0.99	0.45/0.87*
	left of	train	1	1	0.89	0.56/0.56*
		test	0.48	0.4	0.46	0.43/0.43*

* Training is on 1600 samples, testing is on 400.

These results suggest that the TD-BU2 lateral connections have a role in learning the TD specified task and transfer information regarding other possible tasks as well. In the model without TD-BU2 connections, more data is needed to learn a classifier on the specified 'right of' task on feature layer readouts, indicating these connections contribute to the desired task. Moreover, the 'left of' performance of this model deteriorates from the top1 layer to the feature layer, indicating the lateral connections transfer this information too.

4.6 Results on Human-Object Interactions

We present further results on the HICO-DET subset dataset, focusing on tasks 2 (object location to related object location) and 3 (object location to class) where object A is a person, object B is one of the objects described in 4.1.3, and the relation is one of four types of 'riding'. We want to generalize relations to new objects of interaction, e.g. riding a different animal not seen during training. We do this via locations, similar to what was done when generalizing to other objects in section 4.2. We trained task 2 on images of people engaged in riding relations of three types: riding bicycles (type 1), riding animals, elephants and horses (type 2), and riding skateboards, skis and snowboards (type 3). We also train on 'non-riding' (type 4), which includes images of humans interacting with the same objects, but in other relations. These interactions include relations such as 'wash', 'groom' and 'feed'. We tested generalization to images of people riding motorcycles (type 1), cows, giraffes and sheep (type 2), and surfboards (type 3). Specifically, we tested if we could locate new objects in the 'riding' relation and detect cases where the human-object interaction was not 'riding'. In the last step, i.e. task 3, we gave the ground truth location of the target object, e.g. the elephant, and trained to produce the class at this location. This was trained for task 2 train and generalization objects. In sequential testing, the input to task 3 was an approximate location (estimated by task 2), and we checked if it was possible to produce the class near this location.

For this dataset, we represent locations of humans and objects by bounding boxes, i.e. (x,y,w,h), where (x,y) are the coordinates of the top left bounding box corner, and w and h are its width and height. For **task 2**, the input to BU1 is the full image, and the input to the TD is the bounding box of the human of interest in the image. The BU2 has two outputs. The first is a score indicating if the human of interest is 'riding' or not (1 for riding, 0 for non-riding). The second is the offset of the object relative to the human of interest as in InteractNet [20]:

$$\left(\frac{x_o - x_h}{w_h}, \frac{y_o - y_h}{h_h}, log\frac{w_o}{w_h}, log\frac{h_o}{h_h}\right)$$

Where (x_h, y_h, w_h, h_h) represent the human bounding box, and (x_o, y_o, w_o, h_o) the object bounding box. As in InteractNet, we use the Huber loss (similar to MSE loss but more robust to outliers) with $\delta = 1$ (also referred to as the smooth L_1 loss). We used the CS model described in Fig. 2 with resnet-18 for the BU1 layers instead of simple convolution layers.

The following table and figures show that we manage to perform object *localization*. In Tab. 4, each row shows detection results on task 2 for a different 'riding' type, where the last row is their average. The three left columns show results on a test set (of humans riding training objects): the left column shows the mean Intersection over Union (mIoU) of ground truth object bounding boxes and task 2 estimated riding object bounding boxes; the middle shows the percent of test images of each riding type that have mIoU greater than 0.5 (detection); the right shows the percent of images with mIoU greater than 0.1 (localization). Similarly, the three right columns show results on a generalization set (of humans riding generalization objects). Since detection (IoU>0.5) results are not high, we check localization (IoU>0.1) as well. In YOLO [34], the definition for localization is detection with IoU>0.1 and correct classification of objects. Since we separate detection from classification, we refer to localization as achieving IoU>0.1. We checked if this could be sufficient for classification, and discuss this below. In Fig. 7, we show the ROC curves for task 2 'riding' or 'not-riding' predictions for test and generalization sets. Fig. 8 shows some qualitative results on generalization objects.

Table 4: Object detection results on humans riding.

riding type	test			generalization		
	mIoU	IoU>0.5	IoU>0.1	mIoU	IoU>0.5	IoU>0.1
type 1 (bikes)	0.54	0.64	**0.97**	0.51	0.55	**0.98**
type 2 (animals)	0.5	0.57	**0.96**	0.44	0.37	**0.96**
type 3 (boards)	0.22	0.06	**0.72**	0.17	0.02	**0.65**
all	0.39	0.36	**0.86**	0.47	0.48	**0.95**

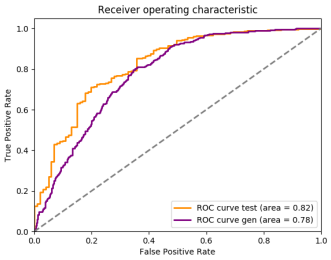

Figure 7: ROC curves for test and generalization 'ride' or 'non-ride' prediction.

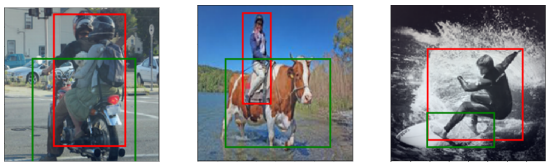

Figure 8: Example generalization images with input human bounding boxes (red) and estimated bounding boxes (green).

Although the detection is far from ideal, we can use the fact that we can localize well enough to perform classification. I.e., perform **task 3**, and achieve classification accuracies on task 2 estimated object locations which are close to ground truth object locations. For this task, we used a resnet-18 BU model, which receives cropped object images as input, and outputs object classes. In training, the object images are cropped by ground truth bounding boxes. In sequential testing, images are cropped by the task 2 estimated bounding boxes, and we want to classify the object at or near this location. This task is learned and tested for all objects, including the generalization objects, which were never seen in the context of the relations, i.e. as a part of task 2. A limitation of this method is that classification is based on cropped images. If estimated object locations are not accurate, it would be difficult to test on images cropped by these estimations. Another option would be to train task 3 with a CS model receiving a full image, instead of a cropped image, and a bounding box TD flag of an object's ground truth or estimated location, and outputting the class via the BU2. The main challenge with this scheme is enforcing the network to use the TD bounding box information and not use the image alone to perform classification.

In Fig. 9 we plot the confusion matrices generated by classifying two versions of this test set, where objects are cropped by ground truth bounding boxes and by the task 2 estimated bounding boxes. The classifier we use (Fig. 9 left) has low accuracy on many object classes, meaning that a main limitation of performance is not generalizing relations to novel objects and their estimated locations, but insufficient training of the classifiers to reach sufficient accuracy. However, the **average recall difference over all classes between ground truth and estimated object bounding boxes is relatively minor**, $12\% \pm 1\%$.

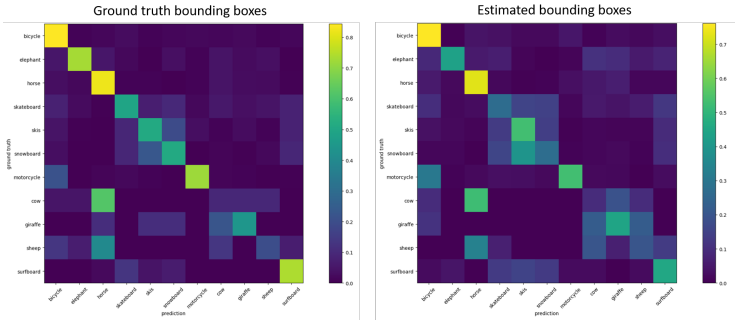

Figure 9: Classification of objects from ground truth bounding boxes (left), and estimated bounding boxes (right).

5 Summary and Conclusion

We have presented in this work a model for relational reasoning, which is explicit, selective and sequential. We showed how to use the BU-TD architecture for relational reasoning and generalization. The proposed architecture has some similarity to the brain's functioning, as far as its TD guidance and the use of lateral connections are concerned. Also similarly to humans, the architecture operates in a sequential manner from object to relation to object. The experiments we described deal with different aspects of two general issues: selection and generalization. **Selection** is the ability to instruct the visual system to perform a particular computation, by applying a particular function to selected item, for example, a particular spatial relation of a selected object in the image. We have shown that such selection can be obtained in computing spatial relations by using a top-down network to instruct and guide the bottom-up network. The selected computations can then be composed sequentially in different combinations to perform more complex computations. The second issue is **generalization**: a major challenge in visual learning is to train on a limited set of examples, and later generalize broadly to different examples. We have shown examples, using both synthetic and natural data, where broad generalization can be obtained by applying a sequence of selected operations.

Using both EMNIST characters and natural images, we showed how it is possible to learn spatial relations regardless of objects, and generalize to different object types, as long as they have similar general statistics (e.g., scale, connectivity, distance, etc.). This is achieved by separating object classification (which can be done on general objects) from learning relations (which can be obtained by learning form a limited set of objects). We also showed it is possible to generalize to new spatial relations, given meaningful relation embeddings. Moreover, we showed we can use multiple cycles to extract more complex information, such as second and third neighbors. As part of this, we introduced the CS model, which is composed of the BU1, TD and BU2 parts, connected by lateral connections. The BU1 and BU2 architectures are the same, and they share the same weights. The BU1 part extracts a full image representation, while the TD part instructs the system to apply a selected computation to a specific object in the image, using the lateral information from BU1. The lateral connections, which are critical for generalization, possibly select and transfer information that is lost in the downsampling from BU1 to TD. It is then used with the TD upsampling layers, guided by a particular selected relation to compute, for attending to a specific area in the image. BU2 classifies specific information extracted through attention, potentially using additional general TD lateral information.

In the future, this work could be used as a building block in a complex system which runs the CS model sequentially, using a sequence of TD instructions, to extract visual relations as well as other information from the image. The appropriate sequence for a task could perhaps be learned by policy learning, using reinforcement learning. If such a policy can output an initial object and relations based on previous CS outputs, this can be used for scene graph and image captioning generation. Such a process could have some similarity to human reasoning during scene understanding, in that it would focus sequentially on different objects and gain more information during each cycle. Such a process may be used for efficient VQA, selecting from the image information needed to produce an answer, and avoiding the need of detecting all objects and relations in an image. Given our results concerning, we expect that generalization to new scene

configurations will depend primarily on having all the appropriate object classifiers for the objects in the scene, but without the need of training the relational reasoning used by the CS model.

References

[1] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[2] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[3] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015.

[4] Victor AF Lamme and Pieter R Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in Neurosciences*, 23(11):571–579, 2000.

[5] Victor AF Lamme, Hans Super, Henk Spekreijse, et al. Feedforward, horizontal, and feedback processing in the visual cortex. *Current Opinion in Neurobiology*, 8(4):529–535, 1998.

[6] André M Bastos, Vladimir Litvak, R Moran, Conrado A Bosman, Pascal Fries, and Karl J Friston. A DCM study of spectral asymmetries in feedforward and feedback connections between visual areas V1 and V4 in the monkey. *Neuroimage*, 108:460–475, 2015.

[7] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 2296–2304, 2015.

[8] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A deep learning approach to visual question answering. *International Journal of Computer Vision*, 125(1-3):110–135, 2017.

[9] Mengye Ren, Ryan Kiros, and Richard Zemel. Image question answering: A visual semantic embedding model and a new dataset. *Proceedings of the Advances in Neural Information Processing Systems*, 1(2):5, 2015.

[10] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[11] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 22–29, 2017.

[12] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, page 2, 2017.

[13] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057, 2015.

[14] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[15] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 3, page 3, 2017.

[16] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 4967–4976, 2017.

[18] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1997, 2017.

[19] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 852–869, 2016.

[20] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[21] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6867–6876, 2018.

[22] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 451–466, 2016.

[23] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[24] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3, 2017.

[25] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross B Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3008–3017, 2017.

[26] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[30] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[31] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

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

[33] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 381–389, 2018.

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.