Relational Reasoning

CS337 Course Project

Karan Godara
210050082

Isha Arora
210050070

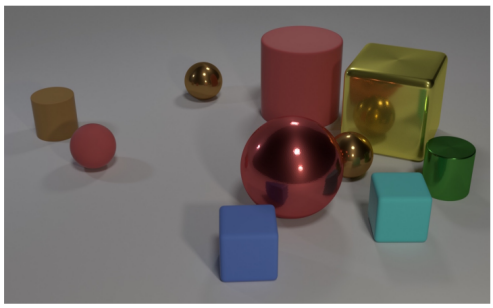
Shantanu Welling
210010076

Dhananjay Raman
210050044

***Abstract*—In this project, we explore the use of Relation Networks in the context of Visual Question Answering. We implement the Relation Network architecture proposed by Santoro et al. [1] and evaluate its performance on the CLEVR dataset [2], as well as a variation of the CLEVR dataset called Sort-of-CLEVR. We try to replicate the results obtained for models corresponding to three different input specifications, namely the input with images and encoded questions, the input with images and unencoded questions, and the input with object descriptions and unencoded questions. We find that Relation Networks perform better compared to state-of-the-art models on both the CLEVR and the Sort-of-CLEVR dataset. We also find that the model performs better when the CNN is replaced with a pre-trained VGG-16 model. We conclude with a discussion on some possible improvements to the model.**

I. Introduction

The ability to reason about the world around us is a fundamental part of human intelligence (Figure 1). It is what allows us to make sense of the world around us, and to make decisions based on our observations. Consider a child proposing a race between the two trees in the park that are furthest apart: the pairwise distances between every tree in the park must be inferred and compared to know where to run. Or, consider a reader piecing together evidence to predict the culprit in a murder-mystery novel: each clue must be considered in its broader context to build a plausible narrative and solve the mystery.



Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

Fig. 1. An example of Relational Reasoning taken from the CLEVR Dataset

A number of approaches to Artifical Intelligence, such as deep learning, struggle in data-poor problems where the underlying structure is characterized by sparse but complex relations [3], [4]. **Our results demonstrate that seemingly simple relational inferences are remarkably difficult for powerful neural network architectures** such as convolutional neural networks (CNNs) and multi-layer perceptrons (MLPs).

Here we explore Relation Networks (RNs) and their application in the field of Relational Reasoning. Relation Networks are a type of neural network architecture that excels at reasoning about complex relationships between objects or entities in a given context. RNs can be used to reason about the relationships between visual elements (such as objects in an image) and textual elements (such as questions about the image) to generate accurate answers.

The key idea behind RNs is to explicitly model pairwise relationships between elements in a given context. This is achieved by using a shared neural network module to process each element individually and then combining the processed representations of each pair of elements to generate a relational representation. This relational representation is then used to make predictions or perform reasoning tasks. Although several other models supporting relation-centric computation have been proposed, such as Graph Neural Networks, Gated Graph Sequence Neural Networks, and Interaction Networks, [5]–[7], RNs are simple, plug-and-play, and are exclusively focused on flexible relational reasoning. Moreover, **through joint training RNs can influence and shape upstream representations in CNNs and LSTMs to produce implicit object-like representations that it can exploit for relational reasoning.**

We applied an RN-augmented architecture to CLEVR [15], a recent visual question answering (QA) dataset on which state-of-the-art approaches have struggled due to the demand for rich relational reasoning. Our networks vastly outperformed the best generally-applicable visual QA architectures, and achieve state-of-the-art, super-human performance. RNs also solve CLEVR from state descriptions, highlighting their versatility in regards to the form of their input. We also show that RNs can be applied to a new dataset, Sort-of-CLEVR, which is designed to test relational reasoning in a simpler setting. Our networks achieve near-perfect accuracy on this dataset, demonstrating that RNs can learn to perform relational reasoning in a data-efficient manner.

II. Relation Networks

A Relation Network (RN) is a specialized neural network module specifically designed for performing relational reasoning tasks. Unlike other neural network architectures, RNs are built with a predefined structure that inherently captures the essential characteristics of relational reasoning. This means that the ability to compute relationships is inherently embedded within the RN architecture, similar to how convolutional neural networks (CNNs) are naturally equipped to reason about spatial properties and recurrent neural networks (RNNs) are designed to handle sequential dependencies.

In its simplest form the RN module is a composite function:

$$RN(O) = f_\phi(\sum_{i,j} g_\theta(o_i, o_j)) \tag{1}$$

where $O = \{o_1, o_2, \ldots, o_n\}$ is a set of n objects ($o_i \in \mathbb{R}^m$), g_θ is a Multi-Layer Perceptron with parameters (learnable weights) θ, f_ϕ is a Multi-Layer Perceptron with parameters ϕ, and $RN(O)$ is the output of the RN module. The function g_θ is used to compute a representation of the relationship between two objects, and f_ϕ is used to aggregate the relationship representations to produce the final output of the RN module. The RN module can be used to perform relational reasoning tasks by feeding it a set of objects and using the output to make predictions or perform reasoning tasks. Since the RN module is a composite function of two MLPs, it can be trained end-to-end using gradient descent.

RNs have **three notable strengths**: they learn to infer relations, they are data efficient, and they operate on a set of objects - a particularly general and versatile input format - in a manner that is order invariant.

A. RNs Learn to Infer Relations

Equation 1 specifies that an RN should take into account all possible relations between object pairs. This means that an RN does not have prior knowledge of which object relations actually exist or their specific meanings. Therefore, RNs need to learn how to infer the presence and implications of object relations. In terms of graph theory, the input can be seen as a complete and directed graph, where the nodes represent objects and the edges represent the object pairs whose relations should be considered. We primarily focus on this "all-to-all" version of the RN in this project.

B. RNs are data efficient

RNs employ a single function, denoted as g_θ, to compute each relation. This function operates on a batch of object pairs, each selected from the same object set. This method fosters superior generalization in relation computation, as g_θ is discouraged from overfitting to specific object pairs. Conversely, a Multilayer Perceptron (MLP) would necessitate embedding n^2 identical functions within its weight parameters to accommodate all possible object pairings, where n is the number of objects. This becomes unmanageable as the number of objects escalates. Hence, RNs provide a more efficient alternative by executing n^2 feedforward passes per object set, considering each possible object pair, and learning the relation function only once.

C. RNs operate on a set of objects

The summation in Equation 1 provides order invariance to the RN's input, respecting the property of sets. This invariance ensures that the RN's output encapsulates information representative of the relations within the object set.

III. Datasets

We applied RN-augmented networks to a variety of tasks that hinge on relational reasoning. To demonstrate the versatility of these networks we chose tasks from the visual question-answering (QA) domain, which requires relational reasoning in a variety of forms. We used the CLEVR dataset [2] to evaluate the performance of our model on a complex relational reasoning task, and the Sort-of-CLEVR dataset to evaluate the performance of our model on a visually simpler relational reasoning task.

A. CLEVR

In visual QA, a model must answer questions about an image, requiring high-level scene understanding and complex relational reasoning over visual and language inputs. However, most visual QA datasets lack fully specified word vocabularies and require comprehensive real-world knowledge. They also contain ambiguities and strong linguistic biases, allowing models to exploit these biases without actually reasoning about the visual input [8]–[10].

Hence, to distill the core challenges of visual QA, the CLEVR dataset was developed [2]. CLEVR contains images of 3D-rendered objects (Figure 2). Each image is associated with a number of questions that fall into different categories. For example, `query attribute` questions may ask "What is the shape of the red object?", while `compare attribute` questions may ask "Is the matte object the same size as the cylinder?".

An important feature of CLEVR is that many questions are explicitly relational in nature. As reported in the original paper, a model comprised of ResNet-101 image embeddings with LSTM question processing and augmented with stacked attention modules vastly outperformed other models at an overall performance of 68.5% (compared to 52.3% for the next best, and 92.6% human performance) [2]. However, for `compare attribute` and `count` questions (i.e., questions heavily involving relations across objects), the model performed little better than the simplest baseline, which answered questions solely based on the probability of answers in the training set.

We used two versions of the CLEVR dataset: (i) the pixel version, in which images were represented in standard 2D pixel form, and (ii) a state description version, in which images were represented by matrices containing object descriptions. Each row in the matrix contained the features of a single object - 3D coordinates (x, y, z); color (r, g, b); shape (cube, cylinder, etc.); material (rubber, metal, etc.); size (small, large, etc.).

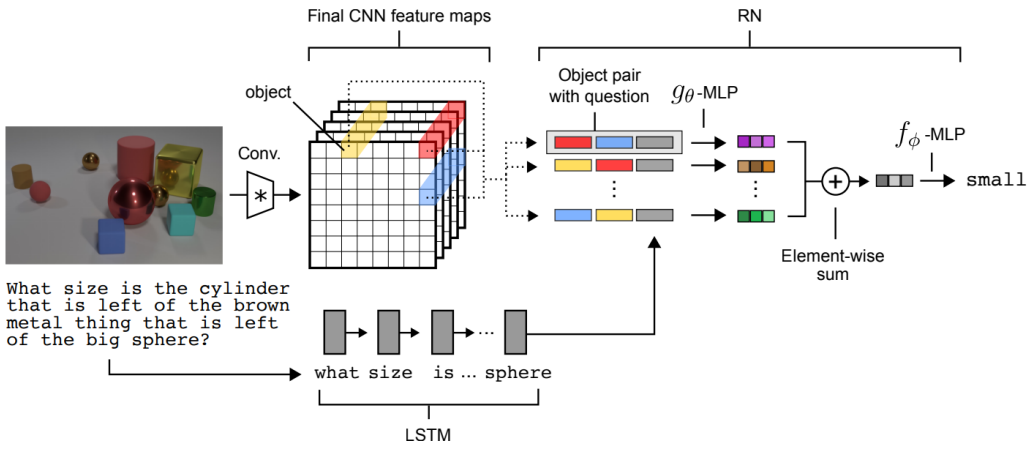


Fig. 2. In Visual QA architecture, questions are processed by an LSTM to generate an embedding, and images are processed by a CNN to create a set of objects for the RN. Objects are formed using feature-map vectors from the convolved image. The RN evaluates relations across all object pairs, based on the question embedding, and combines these relations to provide the answer.

B. Sort-of-CLEVR

This dataset separates relational and non-relational questions. It consists of images of 2D colored shapes along with questions and answers about the images. Each image has a total of 6 objects, where each object is a randomly chosen shape (square or circle). Six colors (red, blue, green, orange, yellow, gray) are used to unambiguously identify each object. Questions are hard-coded as fixed-length binary strings and thereby remove any confounding difficulty with language parsing. For each image, there are 10 relational questions and 10 non-relational questions. Examples of relational questions are: "What is the shape of the object that is farthest from the gray object?". Examples of non-relational questions are: "What is the shape of the gray object?". The dataset is also visually simple, reducing the complexities involved in image processing.

IV. Models

In their simplest form, RNs operate on objects and hence do not explicitly operate on images or natural language. **We demonstrate the flexibility with which relatively unstructured inputs, such as CNN or LSTM embeddings, can be considered as a set of objects for an RN.** Although the RN expects object representations as input, the semantics of what an object is need not be specified. Our results demonstrate that the learning process induces upstream processing, comprising of conventional neural network modules, to produce a set of useful "objects" from distributed representations.

Dealing with pixels: We used a CNN to parse pixel inputs into a set of objects. The CNN took images of size 128 $\times$ 128 and convolved them through four convolutional layers to k feature maps of size $d \times d$, where k is the number of kernels in the final convolutional layer. We remained agnostic as to what particular image features should constitute an object. So, after convolving the image, each of the d^2 k-dimensional cells in the $d \times d$ feature maps was tagged with an arbitrary coordinate indicating its relative spatial position and was treated as an object for the RN (see Figure 2). **This means that an "object" could comprise the background, a particular physical object, a texture, conjunctions of physical objects, etc., which affords the model great flexibility in the learning process.**

Conditioning RNs with question embeddings: The existence and meaning of an object-object relation should be question dependent. For example, if a question asks about a large sphere, then the relations between small cubes are probably irrelevant. So, we modified the RN architecture such that g_θ could condition its processing on the question: $a = f_\phi(\sum_{i,j} g_\theta(o_i, o_j, q))$. To get the question embedding q, we used the final state of an LSTM that processed question words. Question words were assigned unique integers, which were then used to index a learnable lookup table that provided embeddings to the LSTM. At each time-step, the LSTM received a single word embedding as input, according to the syntax of the English-encoded question.

Dealing with state descriptions: We can provide state descriptions directly into the RN, since state descriptions are pre-factored object representations. Question processing can proceed as before: questions pass through an LSTM using a learnable lookup embedding for individual words, and the final state of the LSTM is concatenated to each object-pair.

Model configuration details: For the CLEVR-from-pixels task we used: 4 convolutional layers each with 24 kernels, ReLU non-linearities, and batch normalization; 128 unit LSTM for question processing; 32 unit word-lookup embeddings; four-layer MLP consisting of 256 units per layer with ReLU non-linearities for g_θ; and a three-layer MLP consisting of 256, 256 (with 50% dropout), and 29 units with ReLU non-linearities for f_ϕ. The final layer was a linear layer that produced logits for a softmax over the answer vocabulary. The

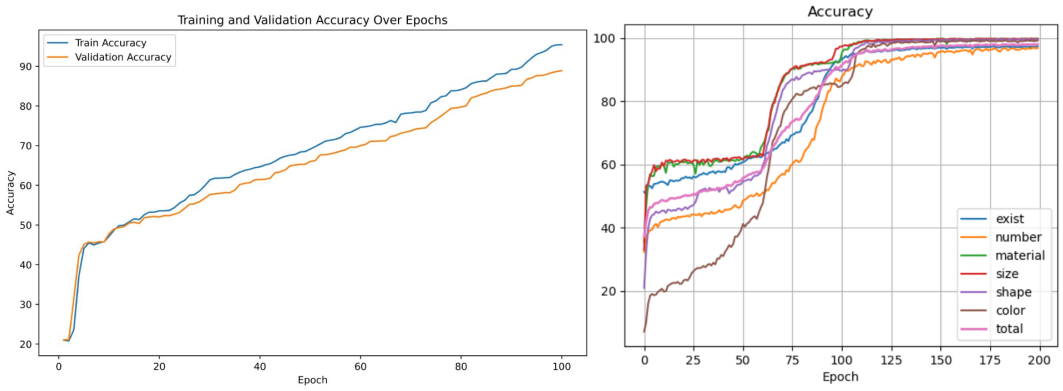


Fig. 3. A comparison of the performance of the CNN+LSTM+RN model (pixel version) and the LSTM+RN model (state description version) on the CLEVR dataset

softmax output was optimized with a negative log-likelihood loss function using the Adam optimizer with a learning rate of 1e-4, with batch size 64.

The model that we train on the state description version of CLEVR is similar to the model trained on the pixel version of CLEVR, but without the vision processing module. The input was a state description matrix of object feature vectors. **This versatility with regards to the definition of object permits the model to deal with state descriptions without the need to take the image scene as input for object processing.** We used a 256 unit LSTM for question processing and word-lookup embeddings of size 32. For the RN we used a four-layer MLP with 512 units per layer, with ReLU non-linearities for g_θ. A three-layer MLP consisting of 512, 1024 (with 2% dropout) and 29 units with ReLU non-linearities was used for f_ϕ. The model was trained with batches of size 640, using the Adam optimizer and a learning rate of 5e-6.

For the sort-of-CLEVR dataset our model used: four convolutional layers with 32, 64, 128 and 256 kernels, ReLU non-linearities, and batch normalization; the questions, which were encoded as fixed-length binary strings, were treated as question embeddings and passed directly to the RN alongside the object pairs; a four-layer MLP consisting of 2000 units per layer with ReLU non-linearities was used for g_θ; and a four-layer MLP consisting of 2000, 1000, 500, and 100 units with ReLU non-linearities used for f_ϕ. An additional final linear layer produced logits for a softmax over the possible answers. The softmax output was optimized with a cross-entropy loss function using the Adam optimizer with a learning rate of 1e-4 and mini-batches of size 64.

We also trained a variant of the model described above, in which the CNN was replaced with a pre-trained VGG-16 model. The VGG-16 model was trained on the ImageNet dataset, and was used to extract features from the images. The extracted features were then passed to the RN.

We also trained a comparable MLP based model (CNN + MLP model) on the Sort-of-CLEVR task, to explore the extent to which a standard model can learn to answer relational questions. We used the same CNN and LSTM, trained end-to-end, as described above. However, this time we replaced the RN with an MLP with the same number of layers and number of units per layer. Note that there are more parameters in this model because the input layer of the MLP connects to the full CNN image embedding.

V. Results

A. CLEVR from pixels

Santoro et al. [1] trained a model which achieved state-of-the-art performance on CLEVR at 95.5%, **vastly outperforming the best model trained only on the pixel images and questions, even surpassing human performance in the task** (Table 1). We were able to replicate these results to an extent, achieving a performance of 88.9%. These results are a testament to the ability of the model to do relational reasoning. Furthermore, **the relative simplicity of the network components used in the model suggests that the difficulty of the CLEVR task lies in its relational reasoning demands, not on the language or the visual processing.**

Model	Accuracy
Human	92.6
Q-type baseline	41.8
LSTM	46.8
CNN+LSTM	52.3
Stacked attention	68.5
CNN+LSTM+RN(Santoro et al.)	95.5
CNN+LSTM+RN(ours)	88.9

TABLE I
Results on CLEVR from pixels. Performances of our models and previously reported models measured as accuracy on the test set

B. CLEVR from state descriptions

To demonstrate that the **RN is robust to the form of its input**, the model was trained on the state description matrix version of the CLEVR dataset. The model trained by Santoro et al. [1] achieved an accuracy of 96.4%, our model achieved an accuracy of 98.1%.

Figure 3 shows a comparison of of the CNN+LSTM+RN model (pixel input) and the LSTM+RN model (state description input) on the CLEVR dataset. It is evident that the LSTM+RN model only performs marginally better.

This result demonstrates the generality of the RN module, showing, because it does not have to extract features from the image, and can instead use the state descriptions directly, **its capacity to learn and reason about object relations while being agnostic to the kind of inputs it receives** - i.e., to the particular representation of the object features to which it has access. Therefore, RNs are not necessarily restricted to visual problems, and can thus be applied in very different contexts, and to different tasks that require relational reasoning.

C. Sort-of-CLEVR from pixels

The model trained by Santoro et al. with a CNN augmented with an RN achieved an accuracy of 94% for both relational and non-relational questions, our model reached 98.7% accuracy on the non-relational questions and 85.6% accuracy on the relational questions as we can see in figure 4.

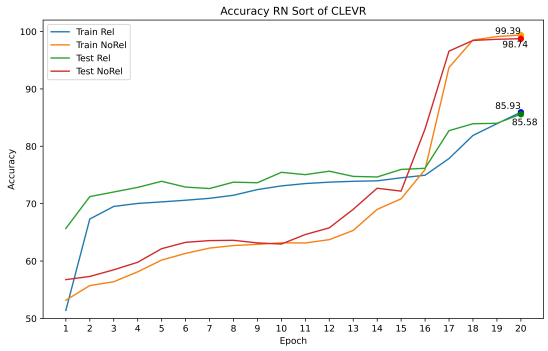


Fig. 4. Performance of RN model on the Sort-of-CLEVR dataset from pixels

In figures 5 and 6, we can observe how there is a high co-relation between the absolute value of gradient in the convolution layer transforming the image to objects and the difference between successive training accuracies. We can observe that there is no such correlation between the difference in the consecutive training accuracies and the absolute gradient of the FC layers which are responsible for finding and using the relation between pair of objects.

The observed phenomenon underscores a **noteworthy relationship between abrupt increases in gradient values and corresponding spikes in accuracy**. This correlation suggests that the model successfully navigated certain local minima due to the use of optimizer Adam, leading to substantial improvements in performance. Notably, this occurrence is particularly pronounced in instances where non-relation questions are involved, **indicating that the final Recurrent Neural Network (RN) model's Fully Connected (FC) layer likely played a minimal role in influencing these notable shifts in non-relational questions.** This also highlights the **important role played by optimizers**, Adam in our case.

Our VGG variant of the RN model reached 98% accuracy on the non-relational questions and 90% accuracy on the

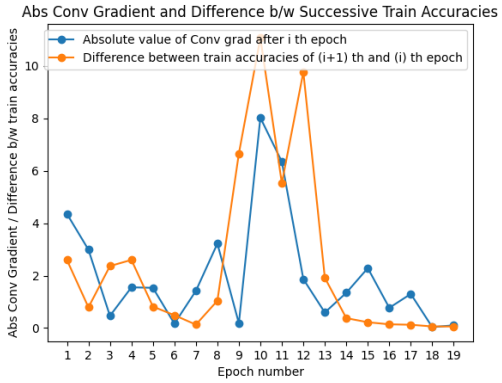


Fig. 5. Absolute Value of the Gradient of Convolution Layer and Difference between Accuracies of Successive Epochs

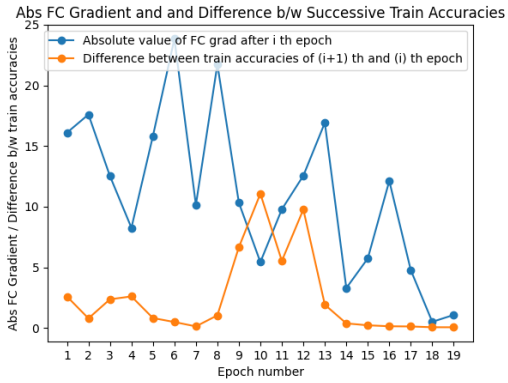


Fig. 6. Absolute Value of the Gradient of Fully Connected Layer and Difference between Accuracies of Successive Epochs

relational questions as we can see in figure 7.This model achieved a better accuracy rate than the previous model. This shows that better the representation of objects, the RN would perform better, and thus **improving what constitutes 'objects' also improves the performance of RN**.

However, as we can see in figure 8 a CNN augmented with an MLP only reached this performance on the non-relational questions, plateauing at 63% on the relational questions. **This strongly indicates that models lacking a dedicated relational reasoning component struggle, or may even be completely incapable of solving tasks that require very simple relational reasoning.** Augmenting these models with a relational module, like the RN, is sufficient to overcome this hurdle.

VI. Discussion and Conclusions

We replicated a pioneering paper on Relation Networks and were able to achieve similar results on both the CLEVR and the Sort-of-CLEVR datasets. We also explored a possible improvement to the model by replacing the CNN with a pre-trained VGG-16 model. We found that the model performed better on the Sort-of-CLEVR dataset when the CNN was

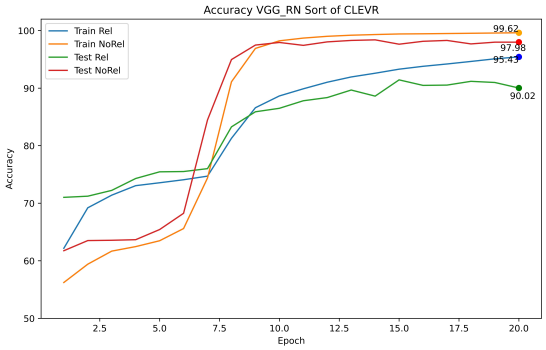


Fig. 7. Performance of our VGG variant of the model on the Sort-of-CLEVR dataset from pixels

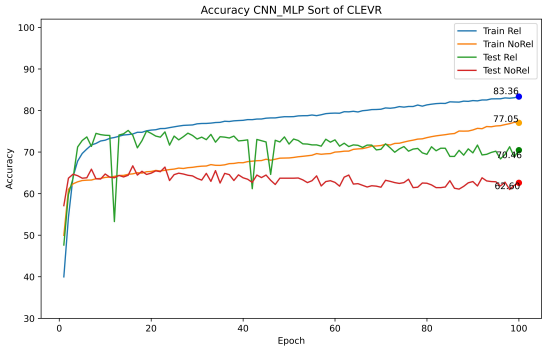


Fig. 8. Performance of the CNN+MLP model on the Sort-of-CLEVR dataset from pixels

replaced with a pre-trained VGG-16 model. This suggests that the model may perform better on the CLEVR dataset as well if the CNN is replaced with a pre-trained VGG-16 model. These results demonstrate the flexibility and power of this simple neural network building block.

One of the most interesting aspects of the work is that **RN module inclusion in relatively simple CNN and LSTM-based VQA architectures drastically raised the performance on the CLEVR dataset and achieved state-of-the-art, super-human performance. We speculate that the RN provided a more powerful mechanism for flexible relational reasoning, and freed up the CNN to focus more exclusively on processing local spatial structure. This distinction between processing and reasoning is important.** Powerful deep learning architectures, such as ResNets, are highly capable visual processors, but they may not be the most appropriate choice for reasoning about arbitrary relations.

A key contribution of this study is that the RN was able to induce, through the learning process, upstream processing to provide a set of useful object-like representations. Note, the input data and target objective functions did not specify any particular form or semantics of the internal object representations. This demonstrates the RN's rich capacity for structured reasoning even with unstructured inputs and outputs.

Possible Improvements Though our results show that no knowledge about the particular relations among objects are necessary, RNs can exploit such knowledge if available or useful. **RN definition can be adjusted to consider only some object pairs**. RNs can take as input a list of only those pairs that should be considered. This information can be explicitly provided in the input data or extracted through some upstream mechanism. For example, if two objects are known to have no actual relation, the RN's computation of their relation can be omitted. An important direction is exercising this option in circumstances with strict computational constraints, where, for instance, **attention mechanisms could be used to filter unimportant relations and thus bound the otherwise quadratic complexity of the number of considered pairwise relations.** In this project, the model was restricted to consider only binary relations between objects. This model could also be extended to take into account **ternary or n-ary relations** between objects.

Relation Networks are a simple and powerful approach for learning to perform rich, structured reasoning in complex, real-world domains.

Acknowledgment

This report was prepared as part of the course CS337: Artifical Intelligence and Machine Learning at IIT Bombay. We would like to thank Prof. Preethi Jyothi of the Department of Computer Science and Engineering, IIT Bombay and all the Teaching Assistants of the CS337 course for their guidance and support though the course of this project.

References

[1] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, T. Lillicrap, "A simple neural network module for relational reasoning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 4967–4976.

[2] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 1988–1997.

[3] M. Garnelo, K. Arulkumaran, M. Shanahan, "Towards deep symbolic reinforcement learning," *arXiv preprint arXiv:1609.05518*, 2016.

[4] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman, "Building machines that learn and think like people," *arXiv preprint arXiv:1604.00289*, 2016.

[5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, January 2009.

[6] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *ICLR*, 2016.

[7] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, K. Kavukcuoglu, et al., "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 4502–4510.

[8] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 2425–2433.

[9] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1–9.

[10] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Advances in Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 2953–2961.