# Maluuba FigureQA: Visual Question Answering for Relational Reasoning

Anish Pimpley Srideepika Jayaraman Shruti Gullapuram Srikanth Grandhe

## Abstract

*The problem of visual question answering deals with coming up with an efficient representation of both the text and visual domains to perform the reasoning task. This is a challenging problem because reasoning in real world requires us to understand how different objects interact and behave with each other in the scene. To build systems that can reason, we need to incorporate concepts such as compositionality, physics, world knowledge etc. which is trivial for humans but not for current intelligent systems. We try to explore this task via the specific problem of question answering in the space of plots and figures using the recently released FigureQA dataset. We build on the ideas of task specific architectures such as Relation Networks and task generic architectures like FiLM to improve the state of the art performance on the FigureQA dataset. We also evaluate our models on the SHAPES, Sort of CLEVR and CLEVR datasets.*

## 1. Introduction

Visual Question Answering is a complex task that requires the knowledge of both visual and textual domains. Performing well on this task requires a strong understanding of relational information between various objects in an image and question provided in textual format. Suppose we are provided with an image and a question such as "Is the ball located next to the bat in the image?", humans may be able to locate the object easily in the image and may find the task to be very trivial. But for an intelligent agent to perform well on this task, the agent in many cases would need multiple sensors to locate the objects in the scene, read the question in textual format, segment the image to locate the objects, construct relationships between the objects and perform reasoning based on the question. Such is the complexity of the visual question answering task enforcing multiple modules to interact to solve this higher order reasoning problem.

In this work, we deal with visual question answering tasks that require relational reasoning. As a solution for this task, we design neural networks that are capable of understanding and performing relational reasoning on different shapes provided in the image. In this context, relational reasoning is analogous to constructing a logical plan with long sequence of reasoning sub-questions defined over objects in the corresponding image.

In order to study this area of work, we look at multiple datasets like SHAPES, CLEVR and Sort of CLEVR that consist of different 2D and 3D shapes such as triangles, squares, cylinders etc. We also evaluate on the FigureQA dataset that aims to apply the reasoning task on scientific figures such plots and graphs.

## 2. Related Work

Visual Question Answering (VQA) was introduced as a free-form and open ended problem domain in 2015 by Agarwal et al.[2]. Being a young domain, research in the area of visual question answering is still in it's nascent stages and no single approach has risen as a clear front runner. Early work in deep learning based models for VQA constituted combining representations from both domains of language and vision in trivial ways such as concatenation, sum etc. to perform reasoning[2].

Our work particularly intends to tackle a specific area of VQA that emphasizes reasoning skills. Models that have performed well on this task, can be broadly classified into a few distinct categories. Namely, Modular, Attention based, Memory based and Relation Nets.

The modular approach adopted by the Program Generator + Executor Engine model [5] obtained competitive results by using a sequence to sequence generator to obtain a tree of compose-able neural network modules, which the Execution Engine uses to predict the answer. The modular approach has its roots in neural module networks [1]. End-to-End Module Networks[3] from the same model family, have also shown promising results on this problem domain. A downside of this approach is that it assumes strong priors in the design of the finite set of modules or requires direct access to ground truth programs.

Stacked attention models[15], Co-attention models[8] and ABC-NN citechen2015abc hail from the attention based approach, where some form of repeated attention is used to highlight relevant parts of the image. The repeatedly attended feature maps have richer representations by encoding information from the other input domain and per-

form well on questions needing multiple steps of reasoning.

Memory based approaches[12][13] have shown promising results in free-form VQA and textual reasoning tasks. Dynamic Memory networks[14][7] borrow from both attention and modular approaches by using the concept of episodic memory with an input module, memory module and an attention mechanism, which is run repeatedly over the question to reason over multiple steps.

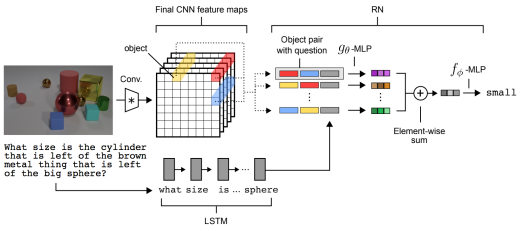


Figure 1. Relation Network model architecture

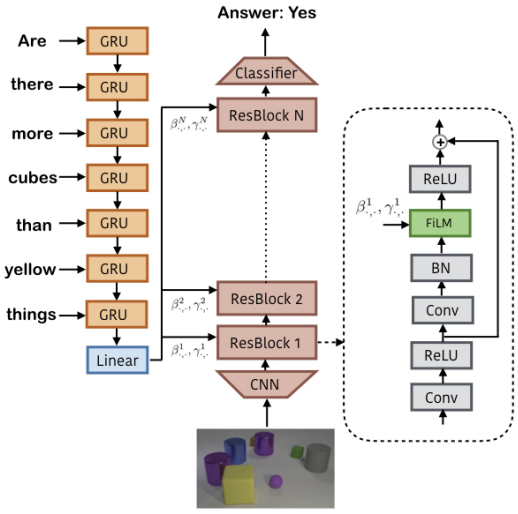


Figure 2. FILM model architecture

Relation Networks[11] were specifically proposed for the spatial relational reasoning task. Relation Networks utilize feature maps and a question embedding obtained from a CNN-LSTM combination, and produce all possible pairwise combinations of spatial positions concatenated with the embedding. All such pairs are then passed through an MLP that encodes the relations between what the authors refer to as objects. A linear combinations of these relations is then passed through an MLP to obtain answers. Relation Networks face computational limitations during both training and testing due to an $O(N^2)$ blowup of possible pairs. Since, Relation Networks compute relationships between object pairs formed from the image features, we explore different architecture choices to overcome some shortcomings of them in our work. There are two divisions of problems that can be addressed with respect to the Relation Networks, the computational efficiency of the model and the performance of the model. Computational efficiency wise, since the RN computes $N^2$ objects for every $N$ features and this increases the time taken, it is possible to reduce the pairs created or reduce the complexity of the model. We however focus more on the performance improvement, by using multiple approaches to incorporate the dependence of the image on the question and vice versa and integrate the models processing on either pipeline.

Recently, FILM[10][9] obtained the state of the art results on CLEVR, by using a generalization conditional batch norm to learn the importance of CNN feature maps at different resolutions in the network conditioned on the question embedding. Conditional batch norm is applied as a channel wise affine transform on CNN residual blocks. One big benefit of FILM, is that the model does not assume any strong priors over the nature of the dataset. The architecture is fairly task agnostic and as of this moment, obtains near perfect results on CLEVR. However, FILM computes parameters of the affine transform in one go, and never uses any visual information in their computation. We see both those areas as avenues for improvement, and have proposed modifications to the architecture targeting those exact modeling decisions.

## 3. Methods

At a high level we wish to explore the problem by building the following class of architectures:

1. Task specific architectures : These architectures embed the reasoning phenomena explicitly into the model architecture allowing the model to perform better on specific reasoning tasks.

2. Task generic architectures : These architectures use the general concepts used in the deep learning community that are task agnostic and perform well on other tasks apart from just reasoning oriented tasks.

### 3.1. Task Specific Architectures

In the task specific architectures, we specifically look to work on the Relation Networks architecture. Relation Networks was used to report the best performance on the FigureQA dataset and we try to improve the performance by modifying the architecture for the same. The following improvements were explored:

1. **Using Attention and Grouping to Reduce Object Features**:
   The benchmark RN model used for FigureQA[6] takes all the feature maps that are generated by feeding the image to a CNN. Hence, if there are 64 feature maps, we say that we have 64 objects to work with. We hypothesized that one way to get robust features while

reducing some redundant relations formed between objects is by grouping them.

We tried to achieve this by using attention mechanisms. The question embedding is used to obtain a weighted sum on a group of features (group size is manually selected i.e. it is a hyperparameter). Thus, if we originally had N objects, and we set our group size to be K, we can reduce the number of comparisons from $\mathbf{N^2}$ to $\frac{\mathbf{N^2}}{\mathbf{K}}$. Figure 1 provides the block diagram for the approach. We propose three variants of group attention namely, group attention with feature interaction (uses question embedding), group attention without feature attention (uses question embedding) and self attention (does not use question embedding).

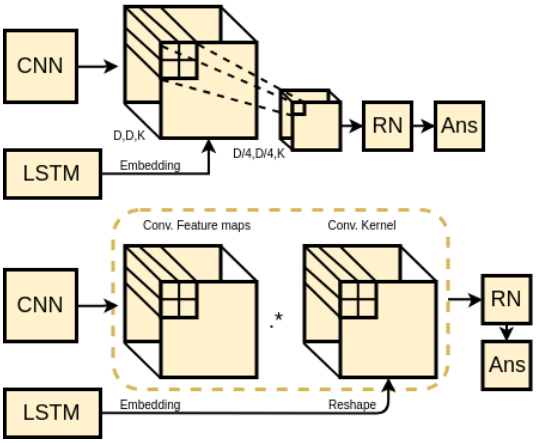


Figure 3. Question Kernel based attention in RN

2. **Convolutional Attention on Image Features**: Another approach we came up with involves conditioning the object features generated by the CNN, based on the question embedding. The question embedding is used to construct a fixed number of kernels that can then be convolved on object feature maps to obtain only the relevant features for the pairwise comparisons. We believe this would help because it introduces question features early in the network and may help to eliminate unnecessary comparisons. Figure 2 provides the block diagram for the approach.

3. **Conditional Batch norm in Relation Networks**: This approach draws inspiration from film[10] where the concept of conditional batch norm is introduced. Since in Relation Networks, the question is only used once the object pairs are constructed, we look to obtain image features that are more relevant to the question. This is done by obtaining batch norm parameters from the question embedding and performing an affine transformation on the output of the image features.

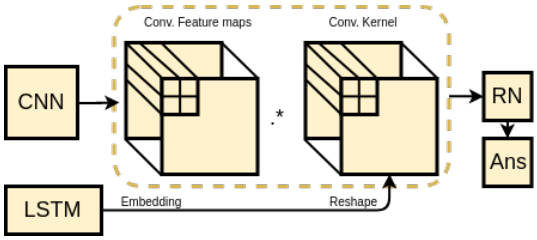


Figure 4. Question Kernel based attention in RN

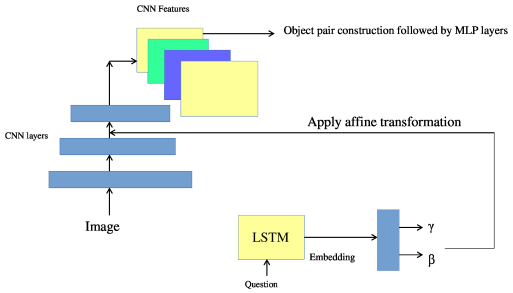


Figure 5. Conditional Batch Norm in RN

## 3.2. Task Generic Architectures

In this section, we explore architectures that don't have a reasoning capability baked into the model architecture itself. With regards to this, we try the following approach:

### 3.2.1 Step-wise Prediction of Conditional Batch Norm Parameters in FiLM

FiLM[10] is a general purpose conditioning method, called Feature wise Linear Modulation. A FiLM layer carries out a simple, feature-wise affine transformation on a neural networks intermediate features, conditioned on an arbitrary input. It is a generalization of Conditional Batch Normalization. The model consists of a linguistic and visual pipeline.

The FiLM model predicts batch norm parameters only based on the question and does not involve the image features during the initial phase. This adds an additional burden on the language model to perform predictions for multiple layers without any information about the image. While the existing architecture is end to end differentiable and performs better on the CLEVR[4] dataset, the architecture by design allows for the propagation of error layer by

layer since all the batch norm parameters are predicted at a single go. This could also mean that the language model has incorporated the biases of the dataset and has been successfully able to map the kind of images that get paired with the questions usually in the dataset.

To overcome this issue, we propose a solution that uses another RNN that takes the question embedding and the layer wise activations iteratively to generate the batch norm parameters. In other words, the input to every Residual Block would be conditioned using the batch norm parameters generated by an RNN which takes both the question embedding as well as the output of the previous ResBlock as inputs. We felt this would also help the model to learn faster as the RNN model will also understand the kinds of features that the CNN generates at each of the layers. Figure 5 shows a block diagram of the approach.

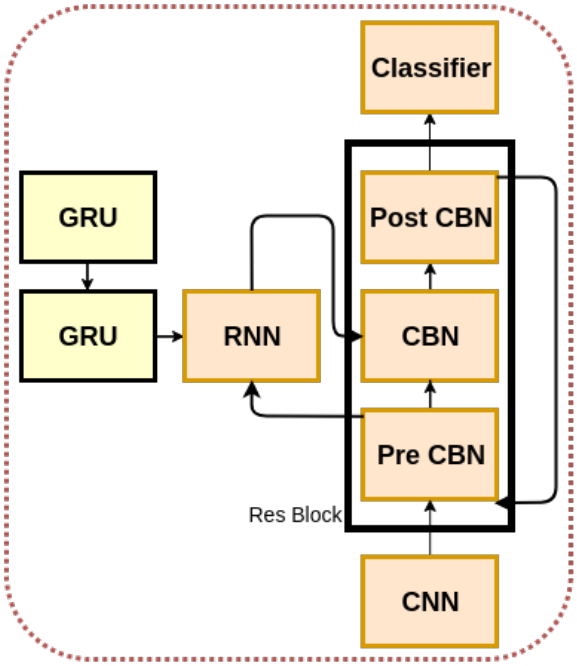


Figure 6. Step-wise prediction of conditional batch norm parameters

## 4. Datasets

To train our models we use the following visual reasoning datasets:

### 4.1. SHAPES Dataset

The SHAPES[5] dataset for visual question answering consists of 15616 image-question pairs with 244 unique questions. Each image consists of shapes of different colors and sizes aligned on a 3 by 3 grid. Questions contain between two and four attributes, object types, or relationships.

### 4.2. CLEVR Dataset

CLEVR[4] is a synthetic dataset which has synthetic images and automatically generated questions. The images have associated ground-truth object locations and attributes, and the questions have an associated machine-readable form. The semantics of these prepositions are complex and depend not only on relative object positions but also on camera viewpoint and context.

### 4.3. Sort-of-CLEVR Dataset

Sort-of-CLEVR is simplified version of CLEVR.This is composed of 10000 images and 20 questions (10 relational questions and 10 non-relational questions) per each image. 6 colors (red, green, blue, orange, gray, yellow) are assigned to randomly chosen shape (square or circle), and placed in a image.

### 4.4. FigureQA Dataset

FigureQA[6] is a visual reasoning corpus of over one million question-answer pairs grounded in over 100,000 images. The images are synthetic, scientific-style figures from five classes: line plots, dot-line plots, vertical and horizontal bar graphs, and pie charts. The questions are generated from 15 templates, concerning various relationships between plot elements and examine characteristics like the maximum, the minimum, area-under-the-curve etc.

## 5. Experiments and Results

In order to conduct our experiments, we test on all the mentioned datasets across all the proposed architectures. For Sort of CLEVER dataset we use an image size of 75 x 75 x 3 while for the rest of the datasets we use input image of size 64 x 64 x 3. For the training purposes, a batch size of 100 was used and the models were trained for 100 epochs. For datasets such as SHAPES and Sort of CLEVR where dataset was not partitioned by default, the train, validation and test set were split in 70%, 20% and 10% proportions respectively. For the training purposes, the popular PyTorch framework was used to construct the models and the models were trained on the TitanX Gpu's.

From table 1, we can get on overview of the how the various architectures perform on different datasets. We can observe that both the baselines CNN + LSTM and Relation Networks perform decently on all the datasets but the suggested improvements do help to boost the performance of the models. On the SHAPES dataset, we observe that the models are overfitting easily and are not able to perform well. This is due to the size of the dataset which too small for the model to learn reasonable patterns from this

dataset. Also the resolution of the images were poor and blurred when up-sampled to 64 x 64 x 3 from 32 x 32 x 3, adding to the usability issue of this dataset for prototyping. In figure 7, we can observe the loss and accuracy plots for the baseline RelNet model and the best performing Conditional batch norm with RelNet model. From these plots we can see that in both the models, the validation loss is constant and does not seem to drop over epochs.

To overcome this issue, we conduct our study on Sort of CLEVR dataset which offers images with desired resolution and is a 10-way classification task including both relational and non-relational questions allowing to analyze the models better. In figure 8, we can observe that the convolutional attention model outperforms the baseline RelNet model and is able to achieve 87.7% accuracy compared to 64.9% achieved by the baseline model. Even the group attention models outperform the RelNet model and are able to achieve around 80%.

On the CLEVR dataset, all the suggested models seem to perform marginally better than the baseline models with the standard group attention models giving the best performance. In figure 9, we can find the accuracy and loss plots for the standard group attention model and the RelNet model.

On the FigureQA dataset, we observe a similar behavior as seen on the CLEVR dataset. The group attention models seems to perform better than the baseline models. For training purposes, we picked a subset of the FigureQA dataset (100K questions) compared to the entire dataset (1M questions) to test since training on the entire dataset is a time consuming process. The results obtained on the sampled dataset may not be representative of the performance of the models on the entire dataset and hence needs to be evaluated on the entire dataset after tuning with proper hyper-parameters. In figure 10, we can observe the loss and accuracy plots of the group attention and the RelNet model. The validation loss plots seem to diverge showing the need for proper hyper-parameter tuning on this dataset.

For obtaining good performance on the group attention models, group sizes of 2, 4 and 8 were used. Tables 3, 4, 5, 6 show the performance of different group attention variants on the SHAPES, Sort of CLEVR , CLEVR and FigureQA datasets respectively. We can see that the group size of 8 gives the best performance on all the datasets. This is due to the fact that we use less convolutional layers and perform attention in the earlier stages compared to the other two group sizes giving more dense features to perform attention.

In table 2, we try to assess the execution times of the different models to find a better and faster performing model. One of the big bottlenecks of the RelNet architecture is the pairwise object comparison resulting in slow model training. We observe that self attention and conditional batch norm models have comparable execution times to the baseline RelNet architecture with comparatively better performance.

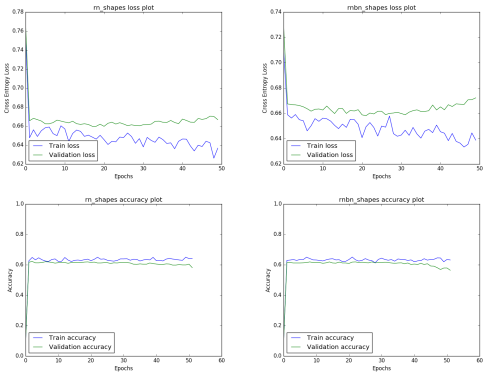


Figure 7. Loss and accuracy plots of baseline and best performing model on SHAPES. (a) Loss plots on top row for RelNet and conditional Batch Norm in RelNet respectively. (Left to Right) (b) Accuracy plots on bottom row for RelNet and Conditional Batch Norm in RelNet respectively (Left to Right)

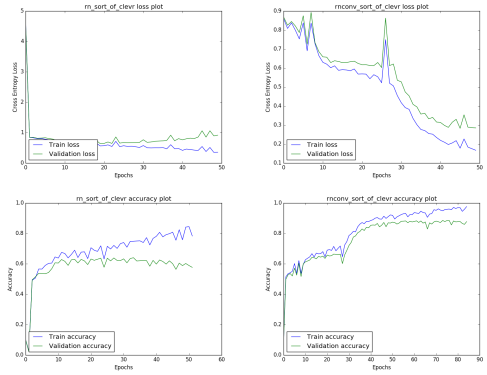


Figure 8. Loss and accuracy plots of baseline and best performing model on Sort of CLEVR dataset. (a) Loss plots on top row for RelNet and Convolutional Attention in RelNet respectively. (Left to Right) (b) Accuracy plots on bottom row for RelNet and Convolutional Attention in RelNet respectively (Left to Right)

## 6. Future Work

In this work, we have explored architectures that are task specific like the RelNet and task generic like the FiLM models and suggested extensions to it. We would like to explore different architecture from scratch. One such model is the stacked co-attention model which incorporates the concept of memory for reasoning. In this model we repeatedly perform attention on the image and question features at each layer of the CNN. The idea is to keep refining the image features based on the question at each layer and then use the attended image features to refine the question embed-

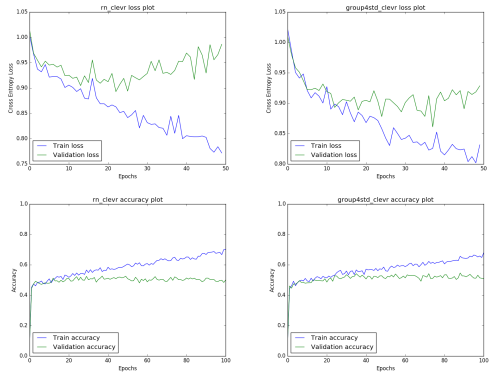


Figure 9. Loss and accuracy plots of baseline and best performing model on CLEVR dataset. (a) Loss plots on top row for RelNet and Group Attention (size 4) in RelNet respectively. (Left to Right) (b) Accuracy plots on bottom row for RelNet and Group Attention (size 4) in RelNet respectively (Left to Right)

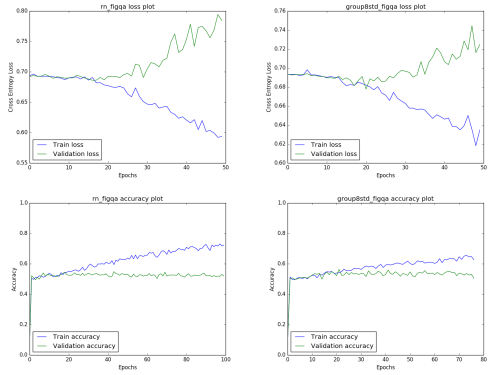


Figure 10. Loss and accuracy plots of baseline and best performing model on FigureQA dataset. (a) Loss plots on top row for RelNet and Group Attention (size 8) in RelNet respectively. (Left to Right) (b) Accuracy plots on bottom row for RelNet and Group Attention (size 8) in RelNet respectively (Left to Right)

ding for the next layer. This model draws idea from the stacked attention network where we apply repeated attention on a single modality and extends it to apply attention on both the modalities. This model can be considered analogous to the end-to-end memory networks. Another viable approach to explore is graph convolutions where instead of just reasoning on pairwise comparisons, we can reasoning over the graph that we construct from the image features conditioned on the questions.

## References


[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016. 1

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1

[3] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. 1

[4] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017. 3, 4

[5] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Inferring and executing programs for visual reasoning. *arXiv preprint arXiv:1705.03633*, 2017. 1, 4

[6] S. E. Kahou, A. Atkinson, V. Michalski, A. Kadar, A. Trischler, and Y. Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017. 2, 4

[7] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016. 2

[8] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016. 1

[9] E. Perez, H. De Vries, F. Strub, V. Dumoulin, and A. Courville. Learning visual reasoning without strong priors. *arXiv preprint arXiv:1707.03017*, 2017. 2

[10] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017. 2, 3

[11] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4974–4983, 2017. 2

[12] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. 2

[13] J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014. 2

[14] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *International Conference on Machine Learning*, pages 2397–2406, 2016. 2

[15] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016. 1

| Model | SHAPES | SORT OF CLEVR | CLEVR | FigureQA (Sampled dataset) |
|---|---|---|---|---|
| CNN + LSTM (Baseline) | 60.1 | N/A | 51.2 | 50.5 |
| Relation Networks | 61.5 | 64.9 | 51.9 | 53.35 |
| Group Attention (Standard) | **62.7** | **79.1** | **54.1** | **56.3** |
| Group Attention (Alternate) | 62.0 | 77.6 | 53.4 | **56.9** |
| Group Attention (Self Attention) | 61.9 | 77.9 | 53.2 | 56.1 |
| Conditional batch norm in RN | **62.5** | 66.4 | 52.4 | 54.9 |
| Convolutional Attention | **62.1** | **87.7** | 53.3 | **55.2** |

Table 1. Overview of performance of different RN models on datasets

| Execution times(seconds/epoch) | CLEVR | FigureQA |
|---|---|---|
| Relation Networks | 1060 | 164 |
| Group Attention (Standard) | 1160 | 260 |
| Group Attention (Alternate) | 1090 | 220 |
| Group Attention (Self Attention) | **990** | 205 |
| Conditional batch norm in RN | **970** | **170** |
| Convolutional Attention | 1800 | 350 |

Table 2. Execution times of models

| Model | Group size 2 | Group size 4 | Group size 8 |
|---|---|---|---|
| Group Attention (standard) | 62.1 | 62.1 | 62.7 |
| Group Attention (alternate) | 61.7 | 61.6 | 62.0 |
| Self Attention | 61.6 | 61.4 | 61.9 |

Table 3. Accuracies of different group attention models vs group size on SHAPES dataset

| Model | Group size 2 | Group size 4 | Group size 8 |
|---|---|---|---|
| Group Attention (standard) | 64.4 | 69.2 | **79.1** |
| Group Attention (alternate) | 63.9 | 68.5 | **77.6** |
| Self Attention | 64.1 | 69.8 | **77.9** |

Table 4. Accuracies of different group attention models vs group size on Sort of CLEVR dataset

| Model | Group size 2 | Group size 4 | Group size 8 |
|---|---|---|---|
| Group Attention (standard) | 51.9 | **54.1** | 53.1 |
| Group Attention (alternate) | 53.3 | **53.4** | **53.4** |
| Self Attention | 51.5 | **53.1** | **53.2** |

Table 5. Accuracies of different group attention models vs group size on CLEVR dataset

| Model | Group size 2 | Group size 4 | Group size 8 |
|---|---|---|---|
| Group Attention (standard) | 54.5 | 55.3 | **56.3** |
| Group Attention (alternate) | 54.4 | 55.4 | **56.9** |
| Self Attention | 53.9 | 55.1 | **56.1** |

Table 6. Accuracies of different group attention models vs group size on sampled FigureQA dataset

# Model evaluation (FigureQA)

| Template (Question Type) | Relation Network (Accuracy) | Group Attention in Relation Network |
|---|---|---|
| Is X the minimum? | 55.8 | **56.23** |
| Is X the maximum? | 51.5 | **54.7** |
| Is X the low median? | 50.3 | **53** |
| Is X the high median? | 49.6 | **52.1** |
| Is X less than Y? | **50.2** | 50 |
| Is X greater than Y? | **50.3** | 46.2 |
| Does X have the minimum area under the curve? | 52.1 | **62.1** |
| Does X intersect Y? | **50.3** | 49.6 |
| Does X have the maximum area under the curve? | 54.9 | **57.2** |
| Is X the smoothest? | 51.7 | **55.2** |
| Is X the roughest? | 52.6 | **58.2** |
| Does X have the lowest value? | 57 | **58.3** |
| Does X have the highest value? | 54.5 | **57.7** |
| Is X less than Y? | 51.6 | **53.6** |
| Is X greater than Y? | 51.6 | **55.3** |

Figure 11. Evaluation of the models on sampled FigureQA dataset