
SARN: Relational Reasoning through Sequential Attention

Jinwon An , Sungwon Lyu, Sungzoon Cho

Department of Industrial Engineering

Seoul National University

Seoul, Republic of Korea

{jinwon, lyusungwon}@dm.snu.ac.kr, zoon@snu.ac.kr

Abstract

This paper proposes an attention module augmented relational network called SARN (Sequential Attention Relational Network) that can carry out relational reasoning by extracting reference objects and making efficient pairing between objects. SARN greatly reduces the computational and memory requirements of the relational network by [6], which computes all object pairs. It also shows high accuracy on the Sort-of-CLEVR dataset compared to other models, especially on relational questions.

1 Introduction

Relational reasoning is one of the fundamental building blocks for all kinds of human cognitive activity [3]. While representing relations and performing reasoning based on them is a challenging problem [5, 2], many graph-based approaches tried to solve these problems [7, 4]. These studies were primarily focused on solving problems using a sparse matrix representing relationships that form a network. Moreover, relational reasoning based on network structure requires clearly defined entities and relations, which is mostly not the case in the real world.

We viewed relational reasoning as a series of decision processes. Consider a general setting where there are many objects in a scene and information on certain relationships has to be inferred based on a given question. The relational reasoning procedure begins with identifying the reference object based on the question. For example, given a certain image to answer the question "What is the closest object near to object A?", we first identify object A, the reference object, in the image. After the reference object is identified, it is compared with other objects to determine the relationship between each pair. Afterwards, distance from object A to other objects will be calculated and sorted to find the nearest object. We do not need the relationship information between object B and object C to answer the question above. In other words, we focus our "attention" only on the relevant relationships by filtering out other relationships that do not include the reference object.

Previous research on relational neural networks that gave intuition to our study was [6] and [8]. [6] proposed the relational module (RN) that tries to compute relational information by pairing each object representation with each other using a relational module. However, because all object pairs were put through the relational module, computational complexity is in $O(n^2)$ where n is the number of objects. [8] used sequentially stacked attention algorithms to focus on areas of images that are relevant to solving a given question. Attention maps enhance performance but also explains how the model views the image when reasoning. Although this incorporates a sequential reasoning process, it does not use an explicit relational reasoning module.

In this work, we propose an efficient relational reasoning algorithm that sequentially processes information using attention. The reference object is found using soft-attention, which is paired with other objects. Next, the relevant relationship between each object and the reference object is extracted.

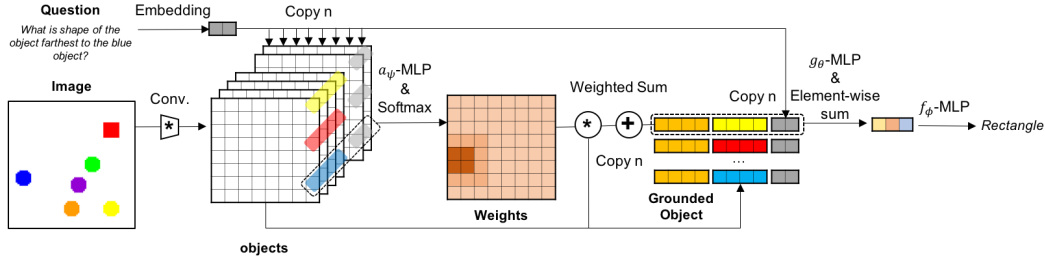


Figure 1: Model architecture overview

By only making object pairs that include the reference object, computational complexity is now in $O(n)$. Attention maps and relational module activation maps show that the results are much more interpretable, selectively showing high activation values in areas of the image that is relevant.

2 Framework and formulation

We use the same notation as in [6]: a convolutional neural network for pixel-wise object representation using feature maps (with coordinate vectors attached), g_θ for the relational module and f_ϕ for processing the aggregated relational information. An additional attention module that we propose for representing the reference object is a_ψ . When we use the word object, we refer to the pixel in the feature map, except for the reference object which is a weighted sum of the pixels.

First, we extract the reference object using soft-attention. The attention module takes the feature map and question embedding as input:

$$a_i = a_\psi(o_i, q) \quad i = 1, \dots, n \quad (1)$$

where o_i is an object and q is the question embedding. It outputs a softmax attention across the objects that locates the reference object. The reference object is represented as a weighted sum of the objects:

$$ro = \sum_{i=1, \dots, n} a_i * o_i \quad (2)$$

The reason why we use soft-attention instead of hard-attention is that each pixel of the feature map does not exactly correspond to one object. In other words, it is possible that the receptive field of a pixel in feature map does not contain an object entirely. It could be distributed among nearby pixels. Selecting only one pixel could force the object representation to be inconclusive. A weighted sum of soft-attention can adequately represent the reference object even in these situations. We check this idea in Section 3.4 by considering various image resolutions.

Next, We pair this reference object representation with other the objects by concatenating it channel-wise. As in a_ψ , the question embedding is concatenated to each pair and is fed to the relational module g_θ :

$$g_{\theta output} = \sum_{i=1, \dots, n} g_\theta(o_i, ro, q) \quad (3)$$

Figure 1 shows the overall model of our proposed model.

3 Experiments

3.1 Dataset

In our experiments, we used a dataset which is a modified version of Sort-of-CLEVR from [6]. Each 75×75 image has 6 objects, whose shape is randomly assigned to be a square or a circle. 6 different colors were used to identify each object. Given a reference object identified by one of the 6 colors, 3

non-relational and 5 relational questions are generated. The non-relational questions are the same as in [6]: (1) horizontal position, (2) vertical position, (3) shape. Relational questions of [6] are (1) shape of the nearest object (2) shape of the furthest object (3) number of objects of the same shape. Additionally, (4) color of the nearest object, (5) color of the furthest object is also added to the relational question list. Sample questions are shown in Figure 2. A total of 9800 images were generated for training and 200 for left for testing. Each image has 48 questions (6×3 non-relational questions, 6×5 relational questions). Question vectors are represented by concatenating two one-hot encoding vectors, one for the color and the other for the question type.

3.2 Models and parameters

We ran three different models. The relational network of [6], a baseline model without object pairing, and our proposed model SARN. Our baseline model is different from that of [6] which flattens out the CNN feature map and concatenate it with the question embedding. We used a different baseline model that takes individual objects as inputs for θ instead of paired inputs and is run through f_ϕ .

The model parameters are the same for each model. CNN: 4 convolutional layers with 32 kernels, ReLU non-linearities, and layer normalization. g_θ , f_ϕ and a_ψ : three-layer MLP with 128 hidden units per layer.

Test accuracy is shown in table Table 1. SARN shows higher accuracy in both non-relational and relational task. For detailed accuracy results for each type of question, see the Appendix. Additionally, the memory and computation time logged by Tensorflow [1] is shown the Appendix.

3.3 Reasoning inspection and interpretability

Since our model runs in a sequential manner, we can examine the attention module and the relational module to verify whether reference objects are correctly retrieved, and important relationships are highlighted.

The attention map produced by a_ψ is shown in Figure 2. It shows the weights of a_ψ . It correctly identifies the region of the reference object according to the question. We did not give only the color embedding vector for a_ψ but used the whole concatenated question embedding vector. a_ψ learned what information to use from the concatenated question embedding vector.

Inspecting the output of g_θ can show whether the most relevant pairs were identified regarding the question. To represent the average activation value for each object-reference pair, g_θ is summed up across channels. This shows the aggregated amount of activation values that each object-reference pair has produced.

Figure 2a shows that a_ψ correctly picks the reference object as the blue object. Regarding channel sum value of g_θ , it also correctly exhibits high activation values for objects that are near the blue object. When the question is finding the farthest objects as in Figure 2 b, high activation values are found near the red object, which is the farthest from the blue object. For other examples, see the Appendix.

We also checked RN if the proper object pairs are used to solve the question. However, it was possible to see that object pairs that do not have much significance in addressing the given question have high activation values of g_θ . This indicates that there is lack of interpretability that can verify the reasoning is done soundly. See the Appendix for detailed examples.

Table 1: Test accuracy

model	overall	non-rel	rel
SARN	96.73	99.84	94.88
RN	93.56	99.81	89.83
base line	89.07	97.58	83.97

3.4 Robustness on image size and object sparsity

We tested how robust SARN is to object size and image size with the same model parameters as in Section 3.2, which are shown in Table 4. By varying image size while fixing object size, we can evaluate how the model deals with sparsity. As image size gets bigger, more and more objects (pixels in the feature maps) will correspond to blank spots. When comparing the configurations where object size and image size are roughly in the same proportion, we can evaluate how the model deals with the granularity of object representation

We first tested robustness to sparsity by varying the image size to 64, 75, 128 while fixing the object size to 5. The baseline model and RN have similar performance on non-relational questions across all image sizes. However, they show worse results on relational questions as image size gets bigger. SARN is relatively robust and even shows higher accuracy with bigger image size.

Next, we tested robustness to granularity by changing the size of image and objects with the same image size (128-4, 128-8). Objects in the configuration (128-8) will be represented by more pixels than in (128-4). In case of RN, this will make each object (pixel) represent only a fraction of the original object in the image. However, SARN takes a soft-attention weighted representation for the reference object and is thus robust to how many pixels represent an original object in the image. The results reflect this: RN shows lower performance as the image size gets bigger. SARN shows stronger performance as the image size gets bigger, especially in relational questions.

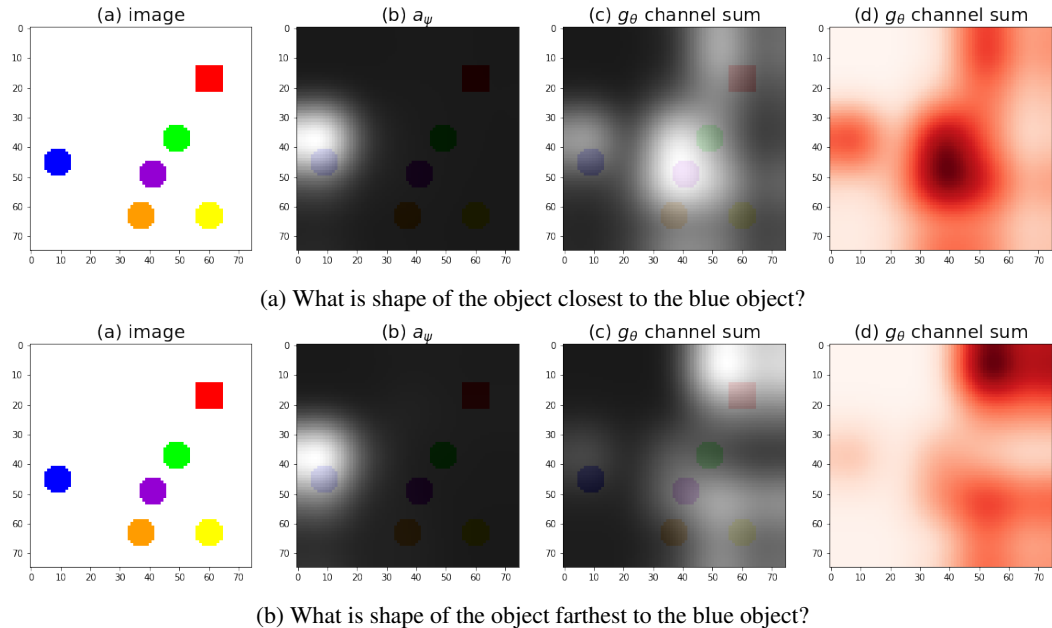


Figure 2: **Sample attention maps and relational module activation:** The first column shows the image. The second column shows the (upscaled) attention map of a_{ψ} overlaid on the image. The third column shows g_{θ} summed up across channels overlaid on the image. The fourth column is used to emphasize and compare the amount of activation value of g_{θ} for each object

4 Conclusion

We propose an attention module augmented relational network called SARN(Sequential Attention Relational Network) that implements an efficient sequential reasoning process of (1) finding the reference object and (2) extracting relevant relationships between the reference object and other objects. This model dramatically reduces the computational and memory requirements of [6], which computes all object pairs. It shows higher accuracy on the modified Sort-of-CLEVR dataset than other models, especially on relational questions. Also by inspecting the attention map and relational module, we can verify that the reasoning process is interpretable. This preliminary research on sequential reasoning shows promising results, which merits further investigation.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [2] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3): 335–346, 1990.
- [3] Charles Kemp and Joshua B Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 2008.
- [4] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.
- [5] Allen Newell. Physical symbol systems. *Cognitive science*, 4(2):135–183, 1980.
- [6] 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 *Advances in neural information processing systems*, pages 4967–4976, 2017.
- [7] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.
- [8] 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.

5 Appendix

5.1 test accuracy by question type

Table 2: Test accuracy: non-relational questions

model	horizontal	vertical	shape	non-rel
SARN	99.92	99.67	99.92	99.84
RN	99.92	99.67	99.83	99.81
base line	96.33	96.58	99.83	97.58

Table 3: Test accuracy: relational questions

model	closest color	closest shape	farthest color	farthest shape	count	rel
SARN	90.75	93.92	93.75	96.33	99.67	94.88
RN	86.33	88.42	84.17	90.25	100	89.83
base line	84.92	88.50	67.83	79.25	99.33	83.97

5.2 Image resolution robustness

Table 4: Accuracy by image size and object size

		64-5	75-5	128-5	128-4	128-8
SARN	non-rel	99.70	99.48	99.88	99.92	99.99
	rel	89.49	83.70	86.69	93.10	94.40
	total	93.45	89.70	91.63	95.66	96.50
RN	non-rel	99.44	99.64	99.31	99.22	99.81
	rel	84.15	84.30	77.19	82.60	82.07
	total	89.89	90.05	85.55	88.83	88.72
baseline	non-rel	99.41	99.33	99.78	98.83	99.72
	rel	81.20	81.63	85.32	72.27	86.25
	total	88.03	88.27	90.74	82.23	91.30

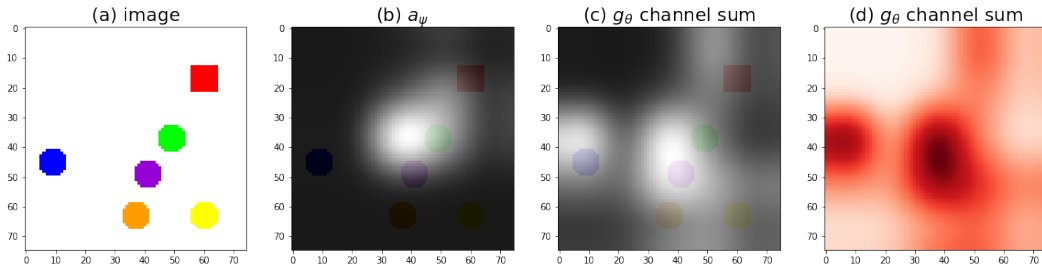
5.3 Memory and computation time in Tensorflow

Table 5 shows the memory and computation time logged by Tensorflow when training a given model varying the input image size. Note that GPU utilization(Nvidia Titan XP) is around 50 60% for SARN and 70 80% for RN. The batch size is 32 for input size 75 and 128. The batch size for input size 256 is 16.

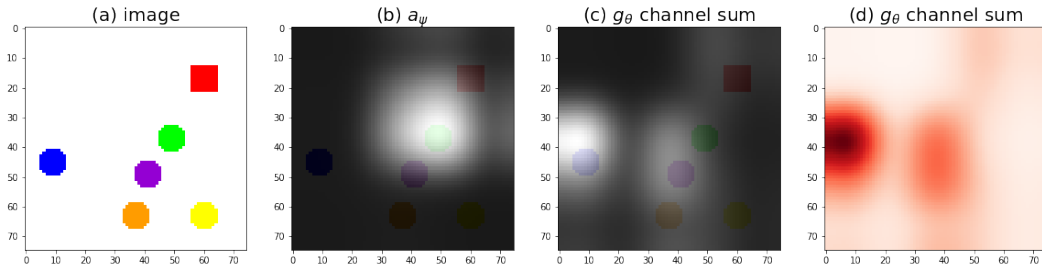
Table 5: Memory and computation time by input size

input size	Model	Computation time(ms)	Memory size(MB)
75	SARN	26.9	142
	RN	24.3	255
128	SARN	30.6	457
	RN	52.5	1580
256	SARN	33.3	414
	RN	117	4650

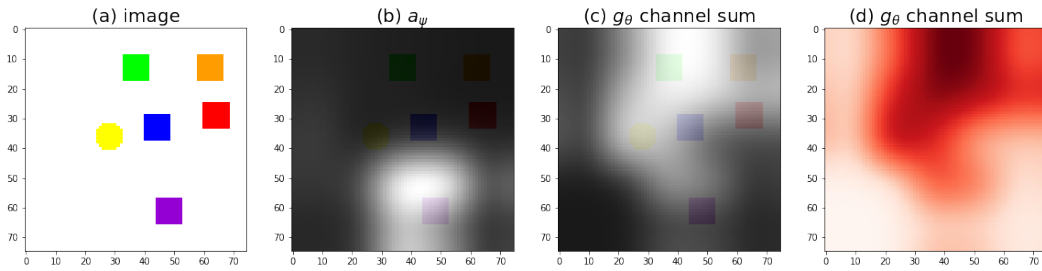
5.4 Proposed model a_ψ and g_θ channel sum plot



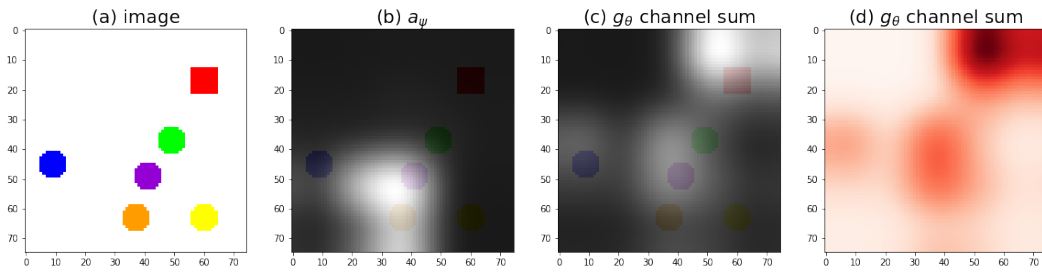
(a) What is shape of the object closest to the green object?



(b) What is shape of the object farthest to the green object?



(c) What is the number of objects that has the same shape as the violet object?



(d) What is color of the object farthest to the orange object?

Figure 3: Sample attention maps and relational module activation: additional examples

5.5 Relational network g_θ channel sum plot for each pair

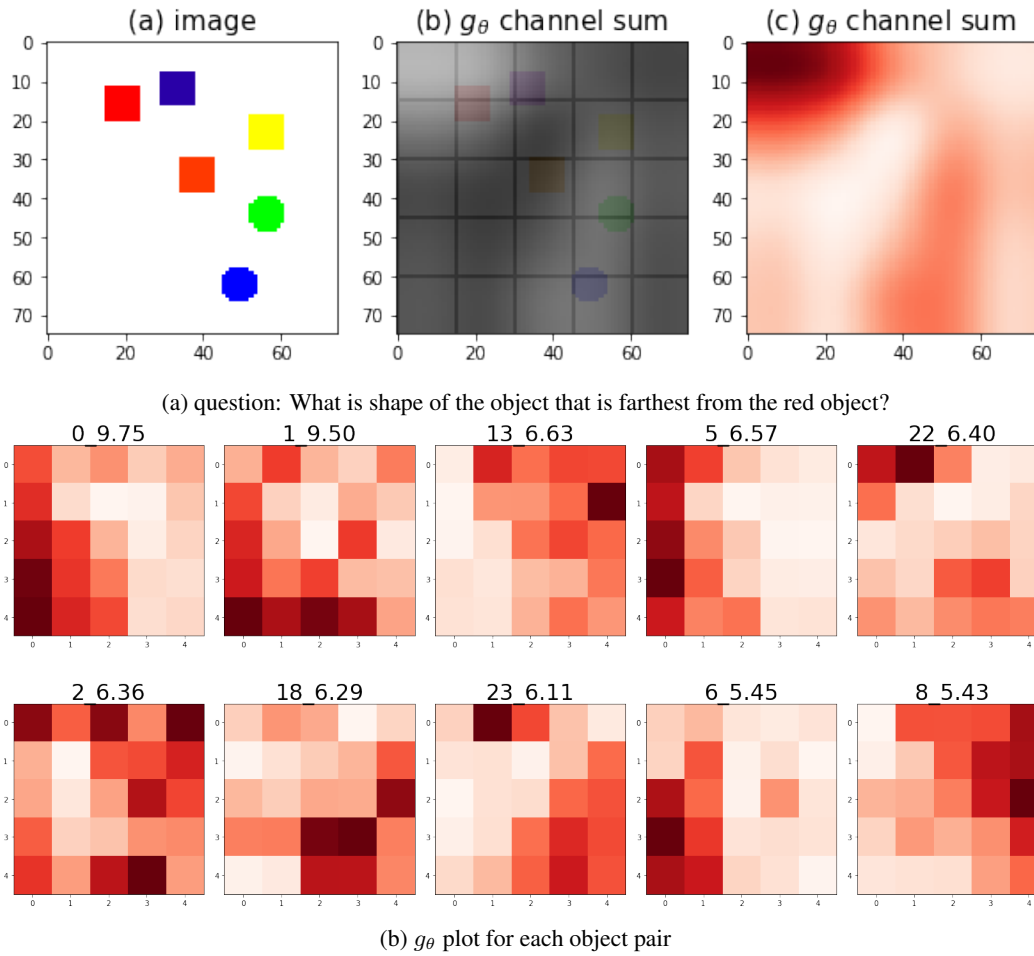


Figure 4: **Relational module activation of RN:** The three figures above show the g_θ output summed up across channels. The 8 figures below show the pairs with the highest activation values. The number above each figure shows the object that is paired with others. The first figure is the g_θ output of object pairs that is paired with the object 0. This tells that objects paired with 0 had the biggest summed up activation value of 9.75. However since the blue object is the furthest from the red object, object 23 should have been the relation that is most critical to solving the problem.