ML Session 14
Relational Reasoning and Relation Networks

Javen Qinfeng Shi

Associate Professor, The University of Adelaide (UoA)

Director and Founder, Probabilistic Graphical Model Group, UoA

Director of Advanced Reasoning and Learning, Australian Institute of Machine Learning (AIML), UoA

Road Map

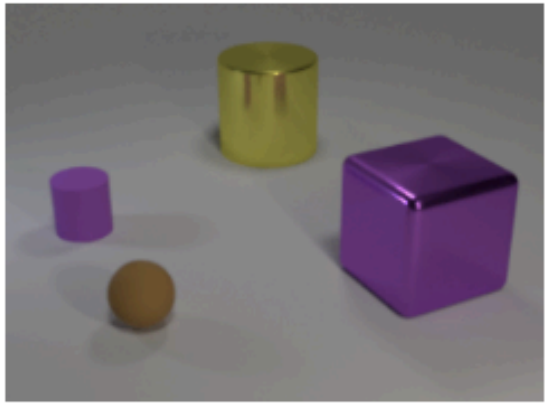
- What is relational reasoning?
- Relation Networks
- Neural-Symbolic VQA

What is relational reasoning?

- To reason or learn the relation between/ among objects.

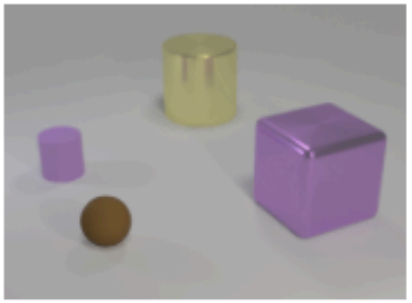
Relational v.s. no-relational

Original Image:



Non-relational question:

What is the size of the brown sphere?



Relational question:

Are there any rubber things that have the same size as the yellow metallic cylinder?

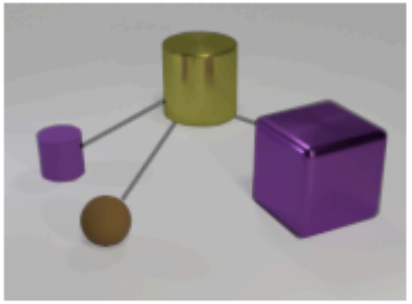


Figure 1: **An illustrative example from the CLEVR dataset of relational reasoning.** An image containing four objects is shown alongside non-relational and relational questions. The relational question requires explicit reasoning about the relations between the four objects in the image, whereas the non-relational question requires reasoning about the attributes of a particular object.

2 main approaches

- Symbolic approach (Symbolism):
  - Define relations between objects using the language of logic and mathematics
  - Interpretable
  - 'Close World' Assumption is often false
  - Can generalize well if careful
  - Less user friendly
- Deep learning approach (Connectivism):
  - Powerful when there are abundant data
  - User friendly
  - Less interpretable
  - Does not generalize well when training on small data

Can we have the benefits of both?

- Yes! A few ways. One of them is:
- Santoro etal NIPS 2018 proposed 'Relation Networks'
  - as a 'general solution to relational reasoning in neural networks'
  - 'RN is a neural network module with a structure primed for relational reasoning'
  - 'the capacity to compute relations is baked into the RN architecture without needing to be learned'

History

- TNN 2009 Graph Neural Networks
- NIPS 2016 Interaction Networks
- NIPS 2017 Relation Networks (cover today)
- NIPS 2018 Neural-Symbolic VQA (cover today)

A simple neural network module for relational reasoning

Adam Santoro,* David Raposo,* David G.T. Barrett, Mateusz Malinowski,
Razvan Pascanu, Peter Battaglia, Timothy Lillicrap

adamsantoro@, draposo@, barrettdavid@, mateuszm@,
razp@, peterbattaglia@, countzero@google.com

DeepMind
London, United Kingdom

RN in the simplest form

$$\mathrm{RN}(O) = f_\phi \left(\sum_{i,j} g_\theta(o_i, o_j) \right)$$

where the input is a set of "objects" $O = \{o_1, o_2, ..., o_n\}$, $o_i \in \mathbb{R}^m$ is the i^{th} object, and f_ϕ and g_θ are functions with parameters ϕ and θ, respectively. For our purposes, f_ϕ and g_θ are MLPs, and the parameters are learnable synaptic weights, making RNs end-to-end differentiable. We call the output of g_θ a "relation"; therefore, the role of g_θ is to infer the ways in which two objects are related, or if they are even related at all.

Notable strengths:

- Learn to infer the existence and implications of object relations
  - Consider all pairwise g_\theta. No need to know which relation exists in advance.
- Operate on a set of objects
  - objects are order invariant.

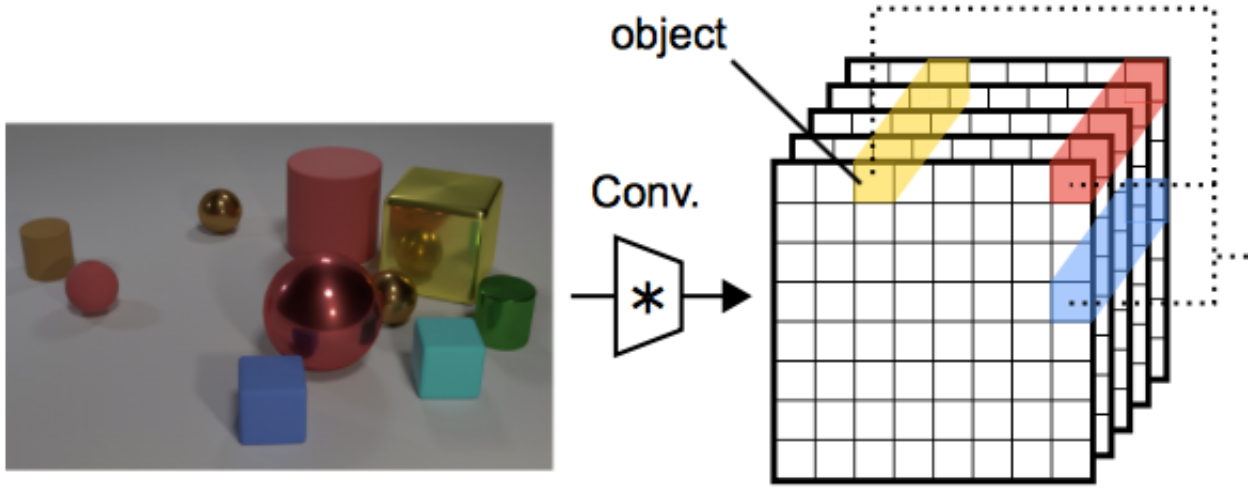
$$\mathrm{RN}(O) = f_\phi \left(\sum_{i,j} g_\theta(o_i, o_j) \right)$$

Tasks

- Visual QA
  - CLEVR dataset
    - pixel version
    - state description version
  - Sort-of-CLEVR dataset
    - Relational questions
    - Non-relational questions
- Text-based QA
  - bAbI pure text based QA dataset. 20 tasks. Each task corresponds to a type of reasoning (like deduction, induction, counting, …)
- Dynamic physical systems
  - Use MuJoCo physics engine to simulate mass-spring system. Each scene has 10 colored balls moving on a table-top surface.
    - Some balls moved independently; Some ball pairs were connected by invisible springs or a rigid constraint.
    - Tasks: 1) to infer the existence of absence of the connection between balls when only observing their color and coordinate positions across multiple frames. 2) Count the number of systems/connected graphs

- RNs (in its simplest form) operate on objects, not images or natural language. How do them tackle VQA and QA tasks?
  - Use CNN or LSTM embeddings as objects

Dealing with pixels We used a CNN to parse pixel inputs into a set of objects. The CNN took images of size 128×128 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

- If ask about a large sphere, then the relations between small cubes are probably irrelevant.
- Modify RN so g_\theta can take question's embedding q as input as well.

$$a = f_\phi(\sum_{i,j} g_\theta(o_i, o_j, q))$$

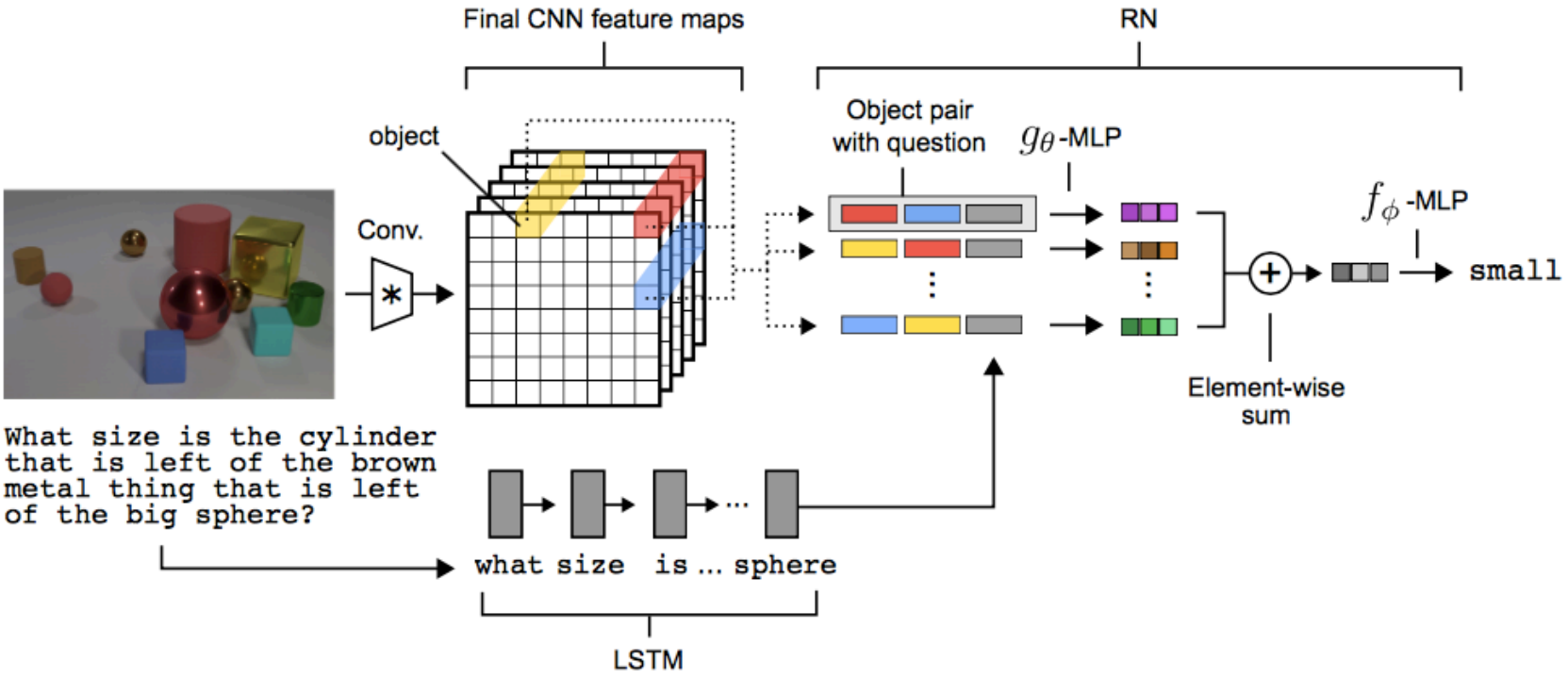


Figure 2: **Visual QA architecture**. Questions are processed with an LSTM to produce a question embedding, and images are processed with a CNN to produce a set of objects for the RN. Objects (three examples illustrated here in yellow, red, and blue) are constructed using feature-map vectors from the convolved image. The RN considers relations across all pairs of objects, conditioned on the question embedding, and integrates all these relations to answer the question.

Model	**Overall**	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
Human	92.6	86.7	96.6	86.5	95.0	96.0
Q-type baseline	41.8	34.6	50.2	51.0	36.0	51.3
LSTM	46.8	41.7	61.1	69.8	36.8	51.8
CNN+LSTM	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA	68.5	52.2	71.1	73.5	85.3	52.3
CNN+LSTM+SA*	76.6	64.4	82.7	77.4	82.6	75.4
CNN+LSTM+RN	**95.5**	**90.1**	**97.8**	**93.6**	**97.9**	**97.1**

* Our implementation, with optimized hyperparameters and trained fully end-to-end.

Table 1: **Results on CLEVR from pixels.** Performances of our model (RN) and previously reported models [16], measured as accuracy on the test set and broken down by question category.

State description version of CLEVR

- State description matrices (without the images)
- Each row describes an object containing
  - 3D coordinates (x,y,z)
  - Color (r,g,b)
  - Shape (cube, cylinder, …)
  - Material (rubber, metal, …)
  - Size (small, large, …)
- Feed state descriptions directly into the RN. Use LSTM embeddings for the questions --- 96.9% overall performance.

Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding

Kexin Yi*
Harvard University

Jiajun Wu*
MIT CSAIL

Chuang Gan
MIT-IBM Watson AI Lab

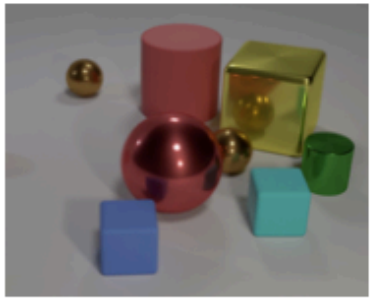
Antonio Torralba
MIT CSAIL

Pushmeet Kohli
DeepMind

Joshua B. Tenenbaum
MIT CSAIL

How many blocks are on the right of the three-level tower?

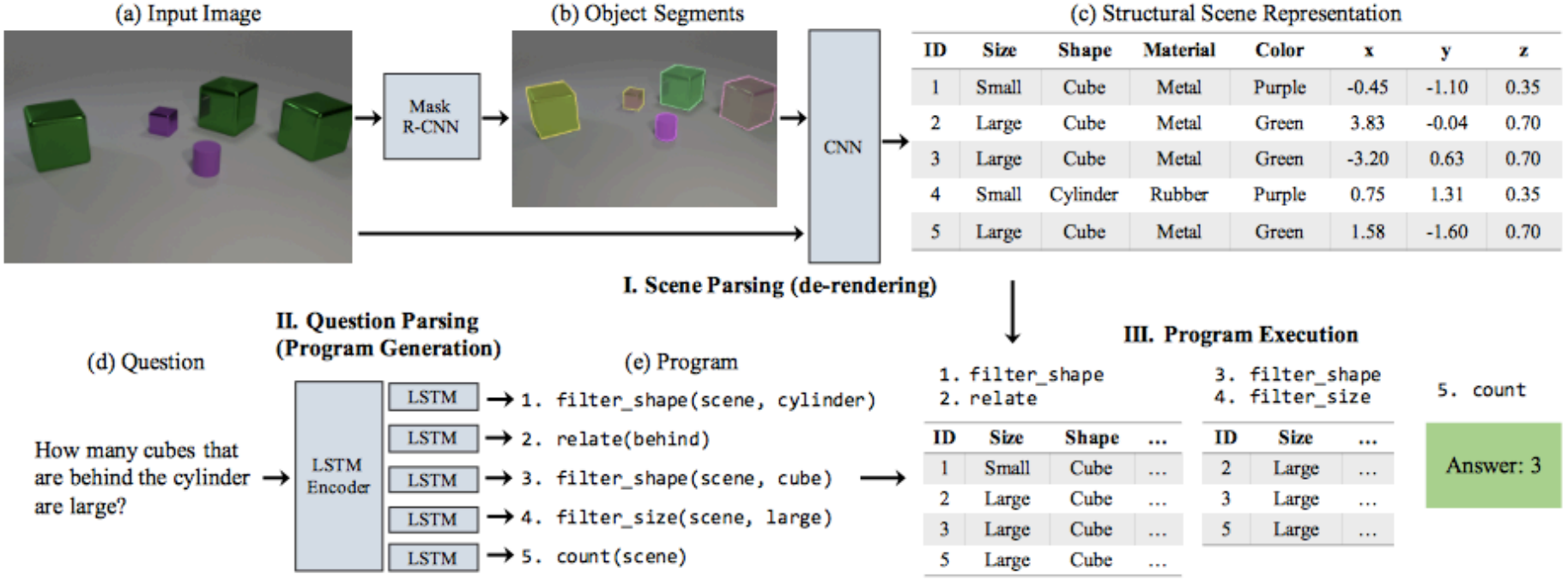
Will the block tower fall if the top block is removed?



What is the shape of the object closest to the large cylinder?

Are there more trees than animals?

Figure 1: Human reasoning is interpretable and disentangled: we first draw abstract knowledge of the scene via visual perception and then perform logic reasoning on it. This enables compositional, accurate, and generalizable reasoning in rich visual contexts.



ID	Size	Shape	Material	Color	x	y	z
1	Small	Cube	Metal	Purple	-0.45	-1.10	0.35
2	Large	Cube	Metal	Green	3.83	-0.04	0.70
3	Large	Cube	Metal	Green	-3.20	0.63	0.70
4	Small	Cylinder	Rubber	Purple	0.75	1.31	0.35
5	Large	Cube	Metal	Green	1.58	-1.60	0.70

ID	Size	Shape	...
1	Small	Cube	...
2	Large	Cube	...
3	Large	Cube	...
5	Large	Cube	...

ID	Size	...
2	Large	...
3	Large	...
5	Large	...

Figure 2: Our model has three components: first, a scene parser (de-renderer) that segments an input image (a-b) and recovers a structural scene representation (c); second, a question parser (program generator) that converts a question in natural language (d) into a program (e); third, a program executor that runs the program on the structural scene representation to obtain the answer.

Methods	Count	Exist	Compare Number	Compare Attribute	Query Attribute	Overall
Humans [Johnson et al., 2017b]	86.7	96.6	86.4	96.0	95.0	92.6
CNN+LSTM+SAN [Johnson et al., 2017b]	59.7	77.9	75.1	70.8	80.9	73.2
N2NMN* [Hu et al., 2017]	68.5	85.7	84.9	88.7	90.0	83.7
Dependency Tree [Cao et al., 2018]	81.4	94.2	81.6	97.1	90.5	89.3
CNN+LSTM+RN [Santoro et al., 2017]	90.1	97.8	93.6	97.1	97.9	95.5
IEP* [Johnson et al., 2017b]	92.7	97.1	98.7	98.9	98.1	96.9
CNN+GRU+FiLM [Perez et al., 2018]	94.5	99.2	93.8	99.0	99.2	97.6
DDRprog* [Suarez et al., 2018]	96.5	98.8	98.4	99.0	99.1	98.3
MAC [Hudson and Manning, 2018]	97.1	99.5	99.1	99.5	99.5	98.9
TbD+reg+hres* [Mascharka et al., 2018]	97.6	99.2	99.4	99.6	99.5	99.1
NS-VQA (ours, 90 programs)	64.5	87.4	53.7	77.4	79.7	74.4
NS-VQA (ours, 180 programs)	85.0	92.9	83.4	90.6	92.2	89.5
NS-VQA (ours, 270 programs)	**99.7**	**99.9**	**99.9**	**99.8**	**99.8**	**99.8**

Table 1: Our model (NS-VQA) outperforms current state-of-the-art methods on CLEVR and achieves near-perfect question answering accuracy. The question-program pairs used for pretraining our model are uniformly drawn from the 90 question families of the dataset: 90, 180, 270 programs correspond to 1, 2, 3 samples from each family respectively. (*): trains on all program annotations (700K).