Date: Feb 5th, 2026

Contents:
Paper Notes and my questions - notes on the paper and questions I had that came up as well as reflections
Experiment Design - based on what I learned from this paper, what can I test to draw my own conclusions?

A simple neural network module for relational reasoning notes

Main point of paper: Relational networks (RN's) are capable of enabling relational reasoning on object detection tasks, and when added onto other powerful neural networks like CNN's (convolutional neural networks) and MLP's (multi-layer perceptrons) can help them gain this capacity. Relation networks are explored in the paper as a **general solution module** to relational reasoning in neural networks in this paper.

Paper's claim: Through joint training, RN's can shape upstream representations in CNN's and LSTM's to make object-like representations.

Question: Upstream vs downstream in machine learning?

Upstream tasks / components /data are those that produce representations, features, or models that others will later use.

Downstream tasks / components are those that consume the representations produced upstream.

Question: What does **joint training** involve?

Instead of training components one after another you optimize them together so they can influence each other during training. The concept of joint training could apply to my research question - see doc *Revelations on Joint Training of Encoder + Neural Network Pipeline* within folder.

An RN in its simplest form is a function that tries to determine the relation between two objects.

RN's learn to infer relations - The form of the equation implies that the RN should consider all pairs involved in the environment and therefore 'calculate' all relations, which means the RN is not pre-aware of which relations can exist and must infer them.

Alternatively, RNs can take as input a list of only the pairs that should be considered which could be explicit in the input data, or could perhaps be extracted by some upstream mechanism.

RNs are data efficient - The one function is used to get the relations between each pair. More efficient than MLP's. Cost of learning a relation function n 2 times using a single feedforward pass per sample, as in an MLP, is replaced by the cost of n 2 feedforward passes per object set (i.e., for each possible object pair in the set) and learning a relation function just once, as in an RN.

Paper Datasets

- RNs were applied to various object detection datasets (CLEVR, Sort-of-CLEVR, bAbI)
    - CLEVR contains images of 3D objects and each image is associated with query attribute questions like "What is the color of the sphere?", while compare attribute questions may ask "Is the cube the same material as the cylinder? ".
    - Two versions of the CLEVR dataset, the pixel version, in which images were represented in standard 2D pixel form, and (ii) a state description version, in which images were explicitly represented by state description matrices containing factored object descriptions.
    - Sort-of-CLEVR is dataset constructed that is like CLEVR but for 2D shapes and separated relational and non-relational dataset questions

My Experiment Design

Below is a short experiment design for me to test that pertains to both the paper and allows me to explore my personal research question.

To evaluate whether relational reasoning requires joint shaping of representation and reasoning mechanisms, I design a minimal analogy task of the form (A:B::C:?), where all examples instantiate a small set of abstract relations (e.g., comparison, containment, role inversion, or ordering) while varying surface realizations. Training examples are drawn from a source domain with a fixed vocabulary and templates, while evaluation is performed zero-shot on a target domain with disjoint surface symbols but identical underlying relational structure. This setting isolates relational generalization from lexical overlap and tests whether learned representations encode relations in a domain-invariant manner.

I compare three architectures: (1) an encoder jointly trained end-to-end with a Relational Network (RN), (2) the same RN trained on top of a frozen encoder pretrained on generic text or symbols, and (3) an encoder-only baseline without an explicit relational module. All models are trained to predict the missing term in the analogy using cross-entropy loss. Performance is measured both on held-out source-domain analogies and on zero-shot target-domain analogies. This comparison directly tests whether joint training is necessary for inducing latent geometries that support systematic relational and analogical reasoning across domains.