Sparse Relational Reasoning with Object-Centric Representations

Alex Spies*, Alessandra Russo, Murray Shanahan

Contact: afspies@imperial.ac.uk *

Paper: bit.ly/sparse_relations

Introduction

For the sake of interpretability and systematic generalization, we want:

- Scene representations which are decomposed into separate objects
- Networks which learn re-usable and simple relations over these.

To this end, we combined:

- Object-Centric Learner - Slot Attention[1]
- Relational Reasoners - RelationNet[2] and PrediNet[3]
- Sparsity Constraints - Object and Feature sparsity

We then investigated the interplay between representations, learned relations and sparsity.

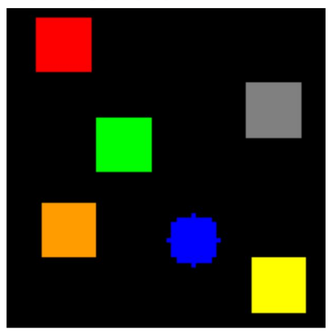
We found that things are not as straightforward as one might hope.

Datasets

Experiments are carried out on two relational reasoning datasets

Relations Game

Sort-of-CLEVR



Two-Stage Curricula:

1. Train entire model on one task (n.b. Slot Attention is pre-trained)
2. Train Task-MLP anew on generalization task (rest frozen)

Multiple tasks per image:

- Non-Relational - "What color is the blue circle?"
- Relational - "What is the color of the square closest to the circle?"

Do Structured Representations Help?

There are many nuances in the results. Broadly, we see that:

- The **inductive bias** present in CNNs is useful when generalizing to tasks which require reasoning over properties not relevant during pretraining (e.g. **y** position differences rather than **x**)
- Slot Attention often **does not capture all objects in a scene**, which can be fatal for tasks requiring knowledge of many objects
- Object-Centric representations can **benefit the RelationNet, but not the PrediNet**. This may simply reflect the poor learning signal for the CNN.

Model		Relations Game			Sort-of-CLEVR	
Encoder	Relational	BC→CP	BC→RM	RM→RP	Non-Relational	Relational
SA	MHA + MLP	56 ± 9	50 ± 2	50 ± 1	85 ± 3	77 ± 3
CNN	PrediNet	**87 ± 3**	**66 ± 0**	74 ± 8	95 ± 1	79 ± 0
GT	Slot PrediNet	**86 ± 2**	**58 ± 2**	**85 ± 2**	85 ± 2	81 ± 1
SA	Slot PrediNet	73 ± 2	53 ± 1	**59 ± 3**	95 ± 2	77 ± 2
CNN	RelationNet	**53 ± 1**	52 ± 1	52 ± 1	100 ± 0	80 ± 1
GT	RelationNet	**75 ± 12**	63 ± 7	55 ± 8	100 ± 0	100 ± 0
SA	RelationNet	70 ± 1	49 ± 1	49 ± 1	100 ± 0	90 ± 1

Does Sparsity Help?

We show results for the BC→CP Relations Game curriculum.

Do Sparsity Priors lead to Sparse Relations?

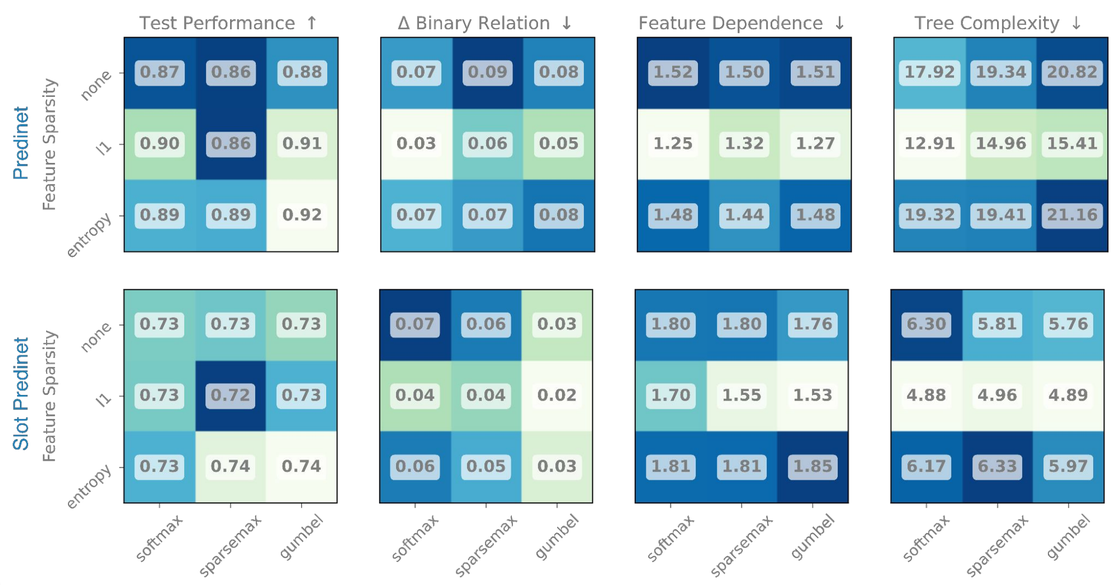
By looking at the Feature Dependence and Tree Complexity we see:

- L1 regularization most effectively regularizes feature dependence
- Gumbel-Softmax is minorly effective, but object sparsity seems negligible

Are Sparser Relations Beneficial?

Looking at test performance and ΔBR:

- Generalization performance is improved by having sparser relations
- Sparser relations do more closely approximate Binary Rules



Model Overview

Architecture

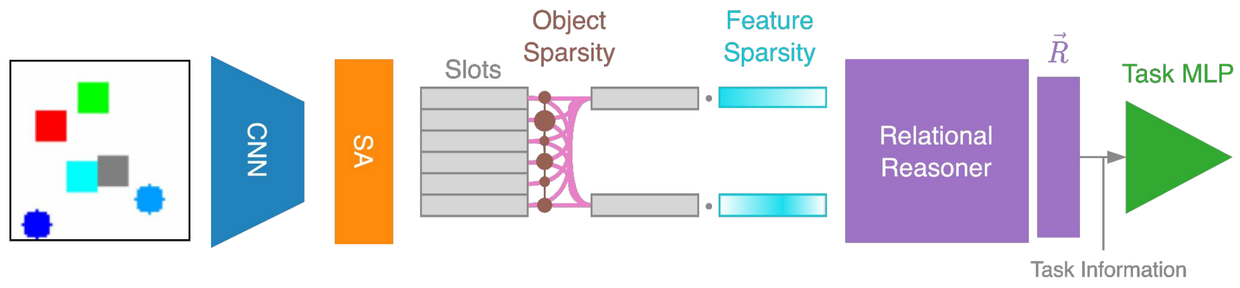
Encoders - All models have a simple 1-layer CNN. To some we add *Slot Attention* which outputs a set of slots that encode objects.

Relational Reasoners - map the output of the encoder to a vector which should encode a set of (binary) relations over the objects in the scene.

- *The RelationNet* applies a shared MLP across all combinations of inputs
- *The PrediNet* applies a set of "heads" which each select a pair of slots using an attention mechanism, subsequently computing relations via learned **W**

Sparsity

- *Object Sparsity* is implemented by using Hard attention in the PrediNet heads (SparseMax[4] and Gumbel-Softmax[5])
- *Feature Sparisty* is implemented by regularizing the learned relations to rely on as few features as possible (L1 and Entropy minimization)



Assessing Sparsity

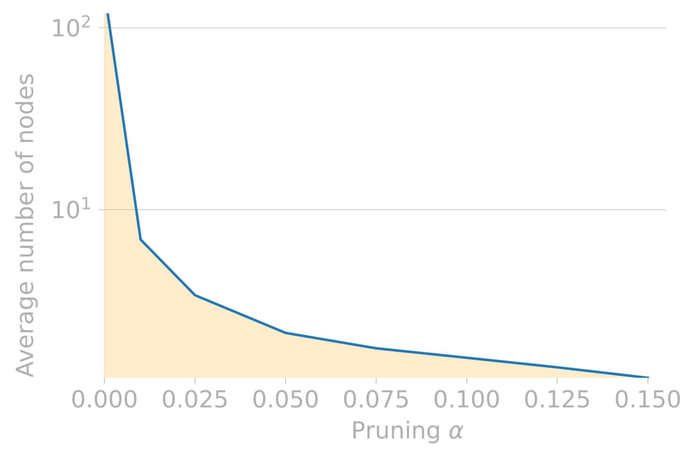
We want to assess how sparse the learned relations are.
To this end we can look at performance, but also at the relations themselves.

We will fit ancillary models (i.e. Decision Trees) to relational reasoners, modelling `input pair → scalar`.

Then we define:

- **Deviation from binary relation** (ΔBR) as the drop in performance when the relational reasoner is replaced with a set of Decision Trees
  → I.e. to what extent can learned relations be modelled as binary rules
- **Feature Dependence** as the average entropy of the "feature importance" of all ancillary models
  → I.e. how sparse are the feature dependencies of learned relations
- **Tree Complexity** as the average size of the Decision Trees over a range of pruning values
  → I.e. how complex are the learned relations

The pruning trades off performance for complexity. As such, these metrics are most informative for models that perform reasonably well in the first place.



Main Findings

Overall, we find that:

- Using Object-Centric representations with Relational Reasoning architectures is **not necessarily beneficial**
- The relations learned by these models, even when they are provided with disentangled inputs, and encouraged toward sparsity, are **not easily interpretable**
- **Object-Sparsity** seems to have little effect on learning relations that generalize well
- **Feature-Sparsity** is beneficial, especially when the relational reasoner is likely to entangle representations (as for the PrediNet)

To address these issues, we need to use/come up with:

- Robust Object-Centric learners which don't undersegment
- Diverse task curricula or ways to re-add lost (spatial) inductive bias
- Ways of enforcing greater sparsity without trading off performance

References

[1] Locatello, F., Weissenborn, D., Unterthiner, T., et al. **Object-Centric Learning with Slot Attention**. NeurIPS 2020

[2] Santoro, A., Raposo, D., Barrett, D. G. T., et al. **A simple neural network module for relational reasoning**. NIPS 2017

[3] Shanahan, M., Nikiforou, K., Creswell, A., et al. **An Explicitly Relational Neural Network Architecture**. ICML 2020

[4] Martins, A. F. T. and Astudillo, R. F. **From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification**. ICML, 2016.

[5] Jang, E., Gu, S., and Poole, B. **Categorical Reparameterization with Gumbel-Softmax.** ICLR, 2017