# Instructions for replication of results

Nicholas Quek

May 25, 2020

## 1 Introduction

This document serves as an instruction manual for replicating the results found in the paper "Pairwise Relations Discriminator for Unsupervised Raven's Progressive Matrices".

# 2 Dependencies

The following is a list of dependencies which we require for the replication:

```
pip install torch
pip install torchvision
pip install scikit-image
```

## 3 Instructions

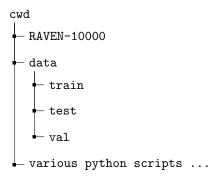
1. Download the RAVEN-10000 dataset [1].

We will be using this dataset of Raven's Progressive Matrices (RPM) for both training and testing of the model. The dataset can be found here  $^{1}$ 

- 2. Unzip the RAVEN-10000 dataset This should produce a directory with seven subfolders, each for a specific configuration. Each subfolder should contain 10,000 RPM problems. Move the directory into the current working directory (cwd).
- 3. Run preprocess.py

```
python3 preprocess.py
```

This python script will preprocess the data in the RAVEN dataset into a format we can easily use. The script will produce a data directory as follows:



<sup>&</sup>lt;sup>1</sup>https://drive.google.com/file/d/111swnEzAY2NfZgeyAhVwQujMjRUfeyuY/view

The train directory will contain 42,000 files named 0 to 41,999. The files can be organised into 7 partitions of 6,000 files. Each partition will correspond to the configurations [Center, 2x2Grid, 3x3Grid, Out-InCenter, Out-InGrid, Left-Right, Up-Down] respectively. i.e. files 6,000 - 11,999 are RPM problems with 2x2Grid configurations; files 24,000 - 29,999 are RPM problems with Out-InGrid configurations.

The test and val directories will each contain 14,000 files named 0 to 1,999. Files are partitioned into blocks of 2,000 and follows the same matching as the partitions in train.

#### 4. Run train.py

```
python3 train.py
```

This script trains the model. We can determine the dataset size by setting the *dataset\_type* hyperparameter. Training will take 200 epochs. The model will take up 4.2 G.B of RAM. Do ensure that your CPU/GPU have sufficient memory.

#### 5. Run validate.py with input of 150

```
python3 validate.py 150
```

This test the model on the validation data at an interval of 5 checkpoints, starting from checkpoint 150. Select the checkpoint that performs the best and modify its filename to *modelA*.

#### 6. Run evaluate.py

```
python3 evaluate.py
```

This loads the model named model A and runs the inference process on the test dataset. The script evaluates the model on the different configurations.

### Notes

Instead of train.py, we can use train\_specific.py to train the model using data from only 1 configuration.

# 4 Results

Table 1: Testing Accuracy of each model on the RAVEN dataset.

		0	•						
Method		Avg	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Supervised	ResNet+DRT LEN + Teacher CoPINet	59.56 78.30 <b>91.42</b>	58.08 82.30 <b>95.05</b>	46.53 58.50 <b>77.45</b>	50.40 64.30 <b>78.85</b>	65.82 87.00 <b>99.10</b>	67.11 85.50 <b>99.65</b>	69.09 88.90 <b>98.50</b>	60.11 81.90 <b>91.35</b>
Unsupervised	Random MCPT PRD	12.50 28.50 <b>50.74</b>	12.50 35.90 <b>74.55</b>	12.50 25.95 <b>38.70</b>	12.50 27.15 <b>34.90</b>	12.50 29.30 <b>60.80</b>	12.50 27.40 <b>60.30</b>	12.50 33.10 <b>62.50</b>	12.50 20.70 <b>23.40</b>
Human		84.41	95.45	81.82	79.55	86.36	81.81	86.36	81.81

## References

[1] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning, 2019.