A. Appendix

A.1. PGM Dataset

Altogether there are 1.2M training set questions, 20K validation set questions, and 200K testing set questions.

When creating the matrices we aimed to use the full Cartesian product $\mathcal{R} \times \mathcal{A}$ for construction structures $\mathcal{S}$. However, some relation-attribute combinations are problematic, such as a progression on line type, and some attributes interact in interesting ways (such as number and position, which are in some sense tied), restricting the type of relations we can apply to these attributes. The final list of relevant relations per attribute type, broken down by object type (shape vs. line) is:

shape:
size: progression, XOR, OR, AND, consistent union
color: progression, XOR, OR, AND, consistent union
number: progression, consistent union
position: XOR, OR, AND
type: progression, XOR, OR, AND, consistent union
line:
color: progression, XOR, OR, AND, consistent union
type: XOR, OR, AND, consistent union

Since the number and position attribute types are tied (for example, having an arithmetic progression on number whilst having an XOR relation on position is not possible), we forbid number and position from co-occurring in the same matrix. Otherwise, all other $((r, o, a), (r, o, a))$ combinations occurred unless specifically controlled for in the generalisation regime.

We created a similar list for possible values for a given attribute:

shape:
color: 10 evenly spaced greyscale intensities in $[0, 1]$
size: 10 scaling factors evenly spaced in $[0, 1]$ [4]
number: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
position ((x, y) coordinates in a (0, 1) plot):
(0.25, 0.75),
(0.75, 0.75),
(0.75, 0.25),
(0.25, 0.25),
(0.5, 0.5),
(0.5, 0.25),
(0.5, 0.75),
(0.25, 0.5),
(0.75, 0.5)
type: circle, triangle, square, pentagon, hexagon, octagon, star

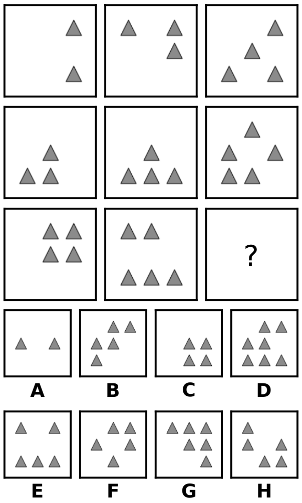
line:
color: 10 evenly spaced greyscale intensity in $[0, 1]$
type: diagonal down, diagonal up, vertical, horizontal, diamond, circle

A.2. Examples of Raven-style PGMs

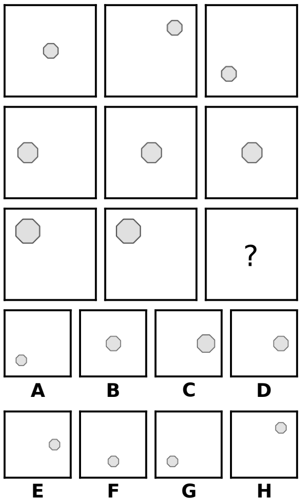
Given the radically different way in which visual reasoning tests are applied to humans (no prior experience) and to our models (controlled training and test splits), we believe it would be misleading to provide a human baseline for our results. However, for a sense of the difficulty of the task, we present here a set of 18 questions generated from the neutral splits. Note that the values are filtered for human readability. In the dataset there are 10 greyscale intensity values for shape and line colour and 10 sizes for each shape. In the following, we restrict to 4 clearly-distinct values for each of these attributes. Best viewed on a digital monitor, zoomed in (see next page). Informal human testing revealed wide variability: participants with a lot of experience with the tests could score well ($> 80\%$), while others who came to the test blind would often fail to answer all the questions.

[4] The actual specific values used for size are numbers particular to the matplotlib implementation of the plots, and hence depend on the scale of the plot and axes, etc.

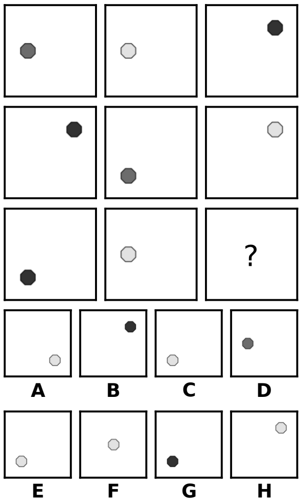
(1)


?
A
B
C
D
E
F
G
H

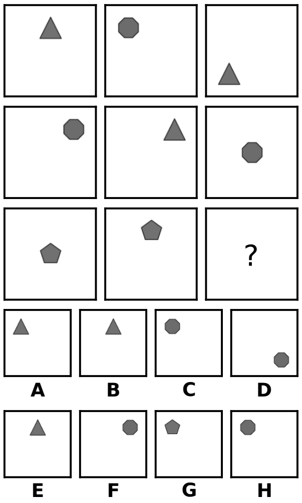
(2)


?
A
B
C
D
E
F
G
H

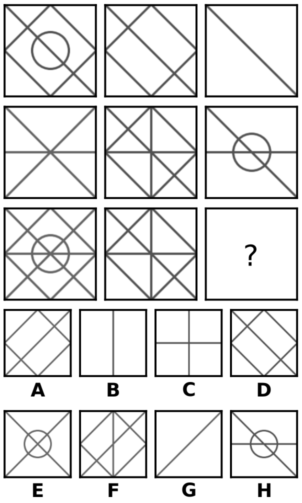
(3)


?
A
B
C
D
E
F
G
H

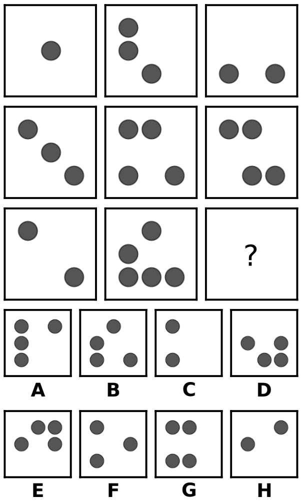
(4)


?
A
B
C
D
E
F
G
H

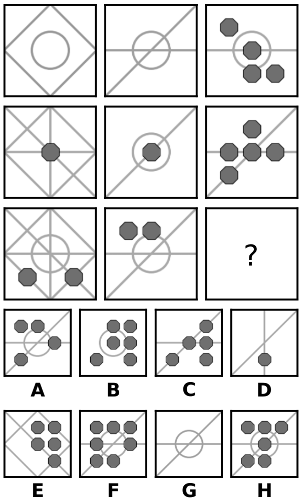
(5)


?
A
B
C
D
E
F
G
H

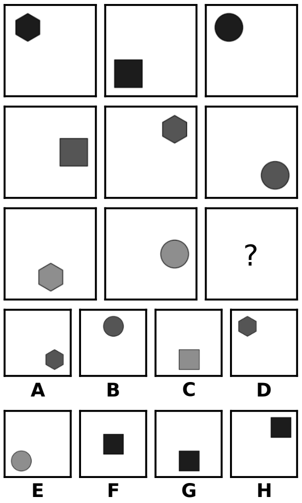
(6)


?
A
B
C
D
E
F
G
H

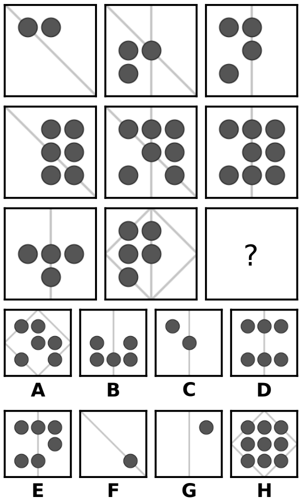
(7)


?
A
B
C
D
E
F
G
H

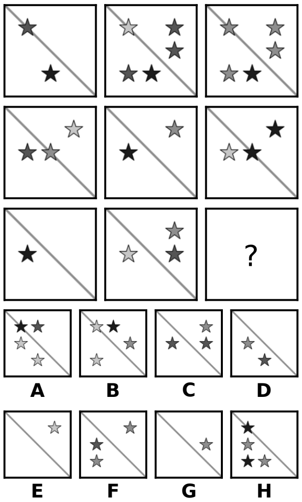
(8)


?
A
B
C
D
E
F
G
H

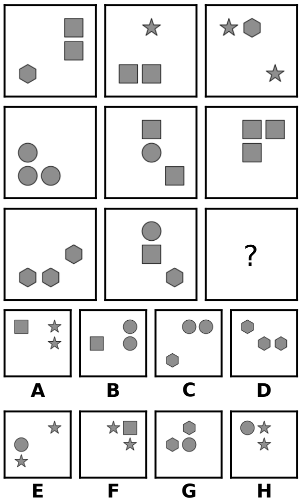
(9)


?
A
B
C
D
E
F
G
H

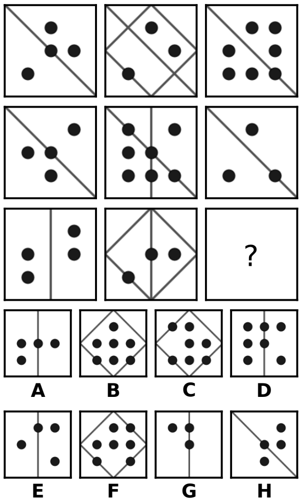
(10)



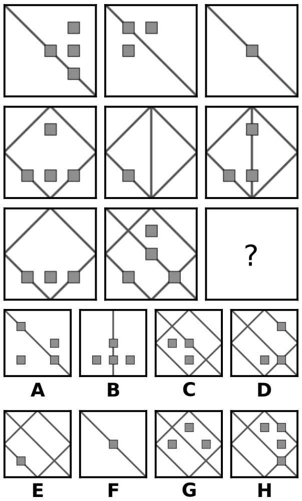
(11)



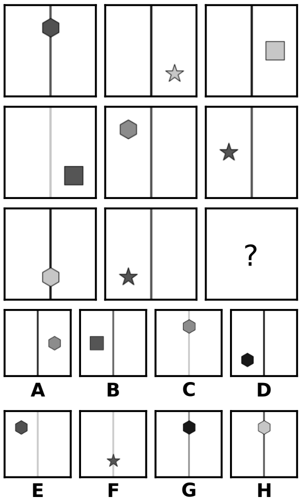
(12)



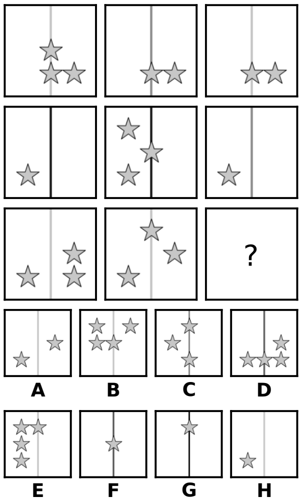
(13)



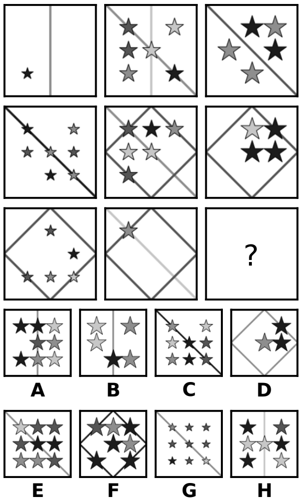
(14)



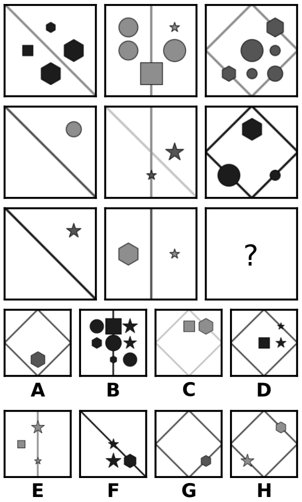
(15)



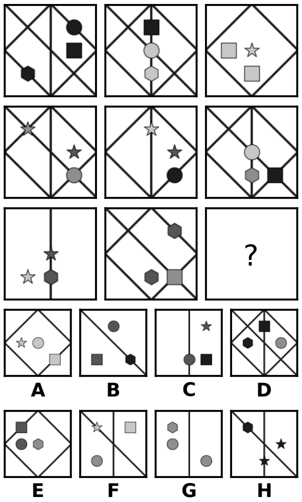
(16)



(17)



(18)



B. Model details

Here we provide additional details for all our models, including the exact hyper-parameter settings that we considered. Throughout this section, we will use the notation $[x, y, z, w]$ to describe CNN and MLP size. For a CNN, this notation refers to the number of kernels per layer: x kernels in the first layer, y kernels in the second layer, z kernels in the third layer and w kernels in the fourth layer. For the MLP, it refers to the number of units per layer: x units in the first layer, y units in the second layer, z units in the third layer and w units in the fourth layer.

All models were trained using the Adam optimiser, with expoential decay rate parameters $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. We also used a distributed training setup, using 4 GPU-workers per model.

	hyper-parameters
CNN kernels	[64, 64, 64, 64]
CNN kernel size	3×3
CNN kernel stride	2
MLP hidden-layer size	1500
MLP drop-out fraction	0.5
Batch Size	16
Learning rate	0.0003

Table 2. CNN-MLP hyper-parameters

	hyper-parameters
Batch Size	32
Learning rate	0.0003

Table 3. ResNet-50 and context-blind ResNet hyper-parameters

	hyper-parameters
CNN kernels	[8, 8, 8, 8]
CNN kernel size	3×3
CNN kernel stride	2
LSTM hidden layer size	96
Drop-out fraction	0.5
Batch Size	16
Learning rate	0.0001

Table 4. LSTM hyper-parameters

	hyper-parameters
CNN kernels	[32, 32, 32, 32]
CNN kernel size	3×3
CNN kernel stride	2
RN embedding size	256
RN g_θ MLP	[512, 512, 512, 512]
RN f_ϕ MLP	[256, 256, 13]
Drop-out fraction	0.5
Batch Size	32
Learning rate	0.0001

Table 5. WReN hyper-parameters

	hyper-parameters
Batch Size	16
Learning rate	0.0003

Table 6. Wild-ResNet hyper-parameters

C. Results

# Relations	WReN (%)	Blind (%)
One	68.5	23.6
Two	51.1	21.2
Three	44.5	22.1
Four	48.4	23.5
All	62.6	22.8

Table 7. WReN test performance and Context-Blind ResNet performance after training on the neutral PGM dataset, broken down according to the number of relations per matrix.

	WReN (%)	Blind (%)
OR	64.7	30.1
AND	63.2	17.2
consistent union	60.1	28.0
progression	55.4	15.7
XOR	53.2	20.2
number	80.1	18.1
position	77.3	27.5
type	61.0	28.1
color	58.9	18.7
size	26.4	16.3
line	78.3	27.5
shape	46.2	18.6
All Single Relations	68.5	23.6

Table 8. WReN test performance and Context-Blind ResNet performance for single-relation PGM questions after training on the neutral PGM dataset, broken down according to the relation type, attribute type and object type in a given matrix.

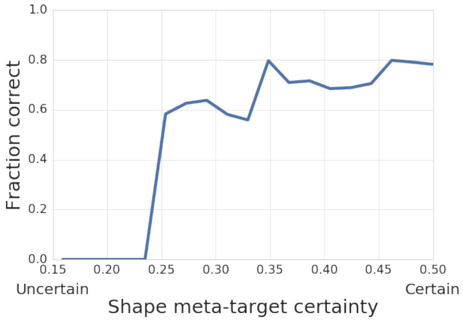


Figure 6. **Relationship between answer accuracy and shape meta-target prediction certainty**. The WReN model ($\beta = 10$) is more accurate when confident about its meta-target predictions. Certainty was defined as the mean absolute difference of the predictions from 0.5.

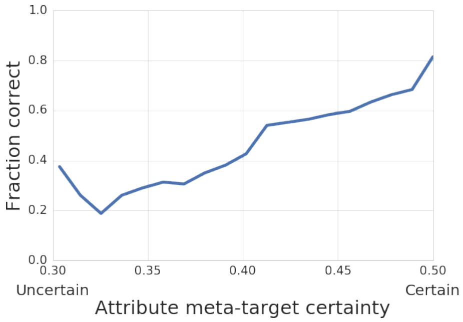


Figure 7. **Relationship between answer accuracy and attribute meta-target prediction certainty**

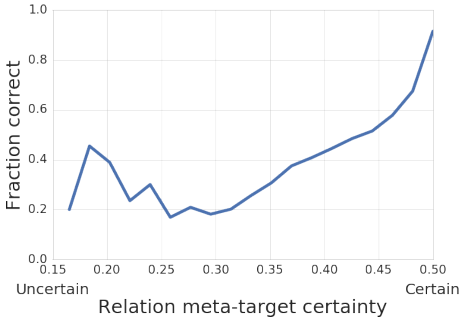
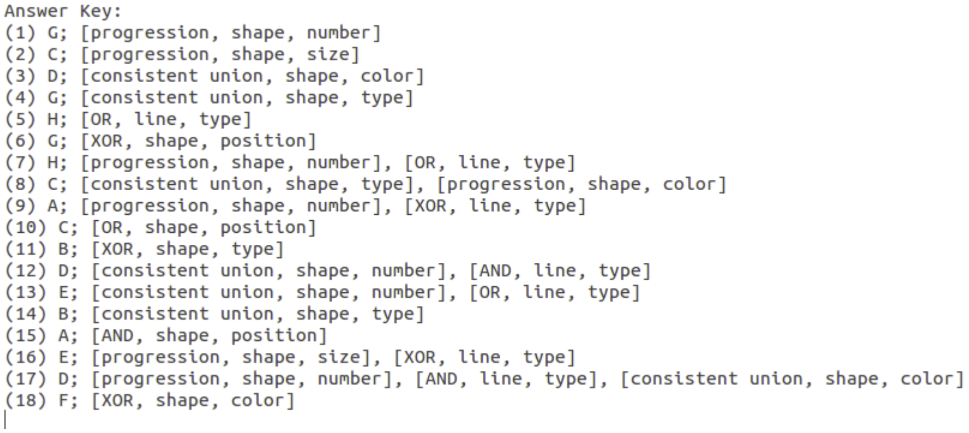


Figure 8. **Relationship between answer accuracy and relation meta-target prediction certainty**

	Test (%)	
Regime	$\beta = \mathbf{0}$	$\beta = \mathbf{10}$
Neutral	22.4	13.5
Interpolation	18.4	12.2
H.O. Attribute Pairs	12.7	12.3
H.O. Triple Pairs	15.0	12.6
H.O. Triples	11.6	12.4
H.O. `line-type`	14.4	12.6
H.O. `shape-colour`	12.5	12.3
Extrapolation	14.1	13.0

Table 9. Performance of the Context-blind Resnet model for all the generalization regimes, in the case where there is an additional auxiliary meta-target ($\beta = 10$) and in the case where there is no auxiliary meta-target ($\beta = 0$). Note that most of these values are either close to chance or slightly above chance, indicating that this baseline model struggles to learn solutions that generalise better than a random guessing solution. For several generalisation regimes such as Interplolation, H.O Attribute Pairs, H.O. Triples and H.O Triple Pairs the generalisation performance of the WReN model reported in Table 1 is far greater than the generalisation performance of our context-blind baseline, indicating that the WReN generalisation cannot be accounted for with a context-blind solution.



```
Answer Key:
(1) G; [progression, shape, number]
(2) C; [progression, shape, size]
(3) D; [consistent union, shape, color]
(4) G; [consistent union, shape, type]
(5) H; [OR, line, type]
(6) G; [XOR, shape, position]
(7) H; [progression, shape, number], [OR, line, type]
(8) C; [consistent union, shape, type], [progression, shape, color]
(9) A; [progression, shape, number], [XOR, line, type]
(10) C; [OR, shape, position]
(11) B; [XOR, shape, type]
(12) D; [consistent union, shape, number], [AND, line, type]
(13) E; [consistent union, shape, number], [OR, line, type]
(14) B; [consistent union, shape, type]
(15) A; [AND, shape, position]
(16) E; [progression, shape, size], [XOR, line, type]
(17) D; [progression, shape, number], [AND, line, type], [consistent union, shape, color]
(18) F; [XOR, shape, color]
```

Figure 9. Answer key to puzzles in section A.2