Yang, Y., McGreggor, K., & Kunda, M. (2020). "Not quite any way you slice it: How different analogical constructions affect Raven's Matrices performance." Proceedings of the Eighth Annual Conference on Advances in Cognitive Systems (ACS). Winner of the inaugural Patrick Henry Winston Award for Best Student Paper on Cognitive Systems.

Not Quite Any Way You Slice It: How Different Analogical Constructions Affect Raven's Matrices Performance

Yuan Yang[1] YUAN.YANG@VANDERBILT.EDU
Keith McGreggor[2] KEITH.MCGREGGOR@GATECH.EDU
Maithilee Kunda[1] MKUNDA@VANDERBILT.EDU
[1]Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA
[2]School of Interactive Computing, Georgia Tech, Atlanta, GA 30308 USA

Abstract

Analogical reasoning fundamentally involves exploiting redundancy in a given task, but there are many different ways an intelligent agent can choose to define and exploit redundancy, often resulting in very different levels of task performance. We explore such variations in analogical reasoning within the domain of geometric matrix reasoning tasks, namely on the Raven's Standard Progressive Matrices intelligence test. We show how different analogical constructions used by the same basic visual-imagery-based computational model—varying only in how they "slice" a matrix problem into parts and do search and optimization within/across these parts—achieve very different levels of test performance, ranging from 13/60 correct all the way up to 57/60 correct. Our findings suggest that the ability to select or build effective high-level analogical constructions can be as important as an agent's competencies in low-level reasoning skills, which raises interesting open questions about the extent to which building the "right" analogies might contribute to individual differences in human matrix reasoning performance, and how intelligent agents might learn to build or select from among different analogical constructions in the first place.

1. Introduction

Raven's Progressive Matrices (RPM) is a very widely-used human intelligence test that contains geometric matrix reasoning problems like those shown in Figure 1—including 2×2 problems (left) and 3×3 problems (right). The task is to select the answer from the options printed at the bottom that best completes the matrix on top.

How do you solve these problems? Your solution process is likely to involve constructing analogies from the problem elements—one row or column becomes the source, another row or column becomes the target, you find a mapping between them, and finally you transfer information from the source to the target to produce an answer—but there are many possible analogies to choose from. For the 2×2 problem on the left, you might construct analogies based on rows or columns. For the 3×3 problem on the right, there are far more variations. Perhaps you just focus on the top row and bottom row, ignoring the middle row completely. Or, maybe you look at the top row first, use the second row to "verify" your hypothesis, and then try to fill in the bottom row.

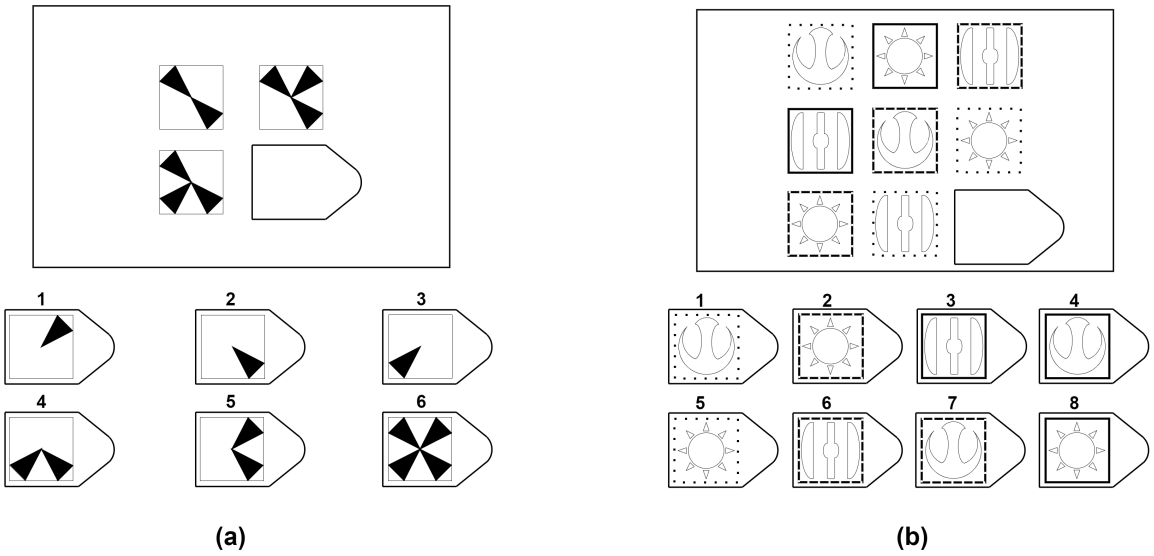

Figure 1. Examples of RPM-like problems: (a) is a 2×2 problem and (b) is a 3×3 problem. Real RPM problems are not shown, to protect the secrecy of the RPM tests.

When taking the RPM, no one tells you how to construct these various analogies to get to the answer. Some research suggests that the ability to construct abstract analogical relations might be an innate ability that distinguishes humans from other species (Hespos et al., 2020). The RPM was specifically designed to test a person's *eductive ability*, or the ability to extract information from and make sense of a complex situation (Raven et al., 1998), where analogies are usually indispensable. Previous computational models have explored many different dimensions of matrix reasoning, including the capacity for subgoaling (Carpenter et al., 1990), specific pattern-matching strategies (Cirillo & Ström, 2010), various forms of rule induction (Rasmussen & Eliasmith, 2011; Little et al., 2012; Shegheva & Goel, 2018), and methods for dynamically re-representing and re-organizing visual elements to find robust mappings (Lovett & Forbus, 2017).

In this paper, we present a systematic examination of another dimension of matrix reasoning, i.e., different ways to construct analogies from matrix elements. As our base model, we use the Affine and Set Transformation Induction (ASTI) model, which operates on scanned, pixel-based images from the RPM test booklet and uses affine transformations and set operations to reason about image differences (Kunda et al., 2013; Kunda, 2013). Our contributions include:

- We present a three-level search hierarchy for representing different types of RPM problem-solving approaches. First, at the level of images, a model can search across a set of **image transformations** to best interpret differences within a given pair or trio of images (e.g., to explain the variation across a given row, column, or diagonal). Second, at the level of a problem matrix, a model can search across different **analogies** to find appropriate transfers of relationships across different pairs or trios of images, including existing rows/columns that appear in the original matrix as well as more complex spatial groupings of elements that can be obtained from spatial re-interpretations of the original matrix. Third, at the highest level of problem solving,

a model can use one of several **integration strategies** that specify how search and optimization at each of the preceding levels are integrated to produce the final answer.

- We demonstrate that a certain combination of transformations, analogies, and integration strategy is sufficient for solving 57/60 problems on the Raven's Standard Progressive Matrices test, which shows that this task-specific language of representations and inference mechanisms is quite expressive with respect to this particular domain.
- Through systematic ablation experiments, we show that test performance can vary widely as a function of overall analogy constructions, i.e., particular selections at different levels of the search hierarchy. For example, if transformations and analogies are held fixed, then variations in integration strategy alone can produce test performance ranging from 13/60 up to 57/60.

2. ASTI+ Model Description

In this section, we describe the expanded Affine and Set Transformation Induction (ASTI) model (Kunda et al., 2013; Kunda, 2013), which we call ASTI+. We describe the model and its variations in terms of five core representations/mechanisms: 1) image representations; 2) similarity metrics; 3) image transformations; 4) matrix analogies; and 5) integration strategies.

2.1 Image Representations

Since the standard RPM is in black and white, we represent each problem as a binary (i.e. pure black and white) image. Hence, an image can also be represented as a set of black pixels. Throughout this paper, we use these two representations interchangeably. Binary images are generated from grayscale scanning images of RPM problems, where a threshold is manually set to convert grayscale values to binary values. An RPM-specific automated image processing pipeline (Kunda, 2013) was used to decompose each full test page into images of individual matrix entries and answer options, as shown in Figure 2. These individual images are then fed as inputs to the ASTI+ models.

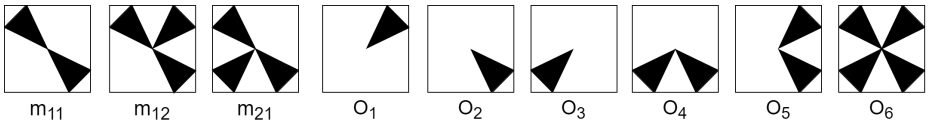

Figure 2. Illustration of input to our models for the 2×2 example problem: m_{ij} is the matrix entry in Row i and Column j, and O_k is the k-th answer option.

2.2 Similarity Metrics

We use the Jaccard index and the asymmetric Jaccard index to measure the similarity between images, as shown in Equation 1 and 2, where A and B are two sets representing two binary images. Equation 2 is asymmetric because $J_A(A, B) \neq J_A(B, A)$, and it measures the extent to which A is inside (or a subset of) B.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

$$J_A(A, B) = \frac{|A \cap B|}{|A|} \tag{2}$$

A problem in Equation 1 and 2 is that A and B should be well aligned. In other words, A and B should have the same shape and size, and pixels at the same coordinates in A and B should spatially correspond. However, the images of individual matrix entries and options come in various shapes and sizes.

We take a simple and robust approach to this problem — slide one image over the other (like a correlation filter), calculate a similarity value at every relative position, and select the maximum similarity. In the process of sliding, images are padded to have the same shape and size in order for them to be fed into Equation 1 and 2.

$$S(A, B) = (J(A, B), pos_{AB}) \tag{3}$$

$$S_A(A, B) = (J_A(A, B), pos_{AB}, pos_{DA}, D) \tag{4}$$

As a result, similarity procedures in our models are defined in Equation 3 and 4, where $J(A, B)$ and $J_A(A, B)$ are the maximum similarity values at the relative position pos_{AB} of A to B, $D = B - A$ aligned by pos_{AB}, and pos_{DA} is the relative position of D to A.

2.3 Transformations

The ASTI+ models represent low-level visuospatial domain knowledge in the form of a discrete set of image transformations, i.e. functions that map from one or more input images to an output image. All of these functions operate on images at the pixel level, without re-representing visual information in terms of higher-order features, shapes, etc. While these functions were defined manually, based largely on inspections of the Raven's test, important directions for future work include expanding these functions to include higher-order features and concepts, as well as learning these functions from perceptual experience (Memisevic & Hinton, 2010; Michelson et al., 2019).

There are two types of ASTI+ image transformations: unary and binary. Unary transformations take a single input image, while binary transformations take two input images. All of the ASTI+

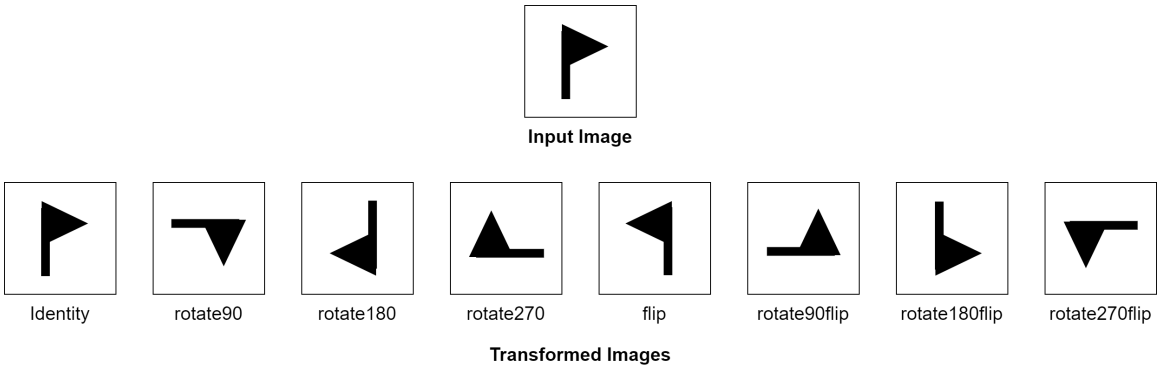

Figure 3. Illustration of affine transformations used in our models.

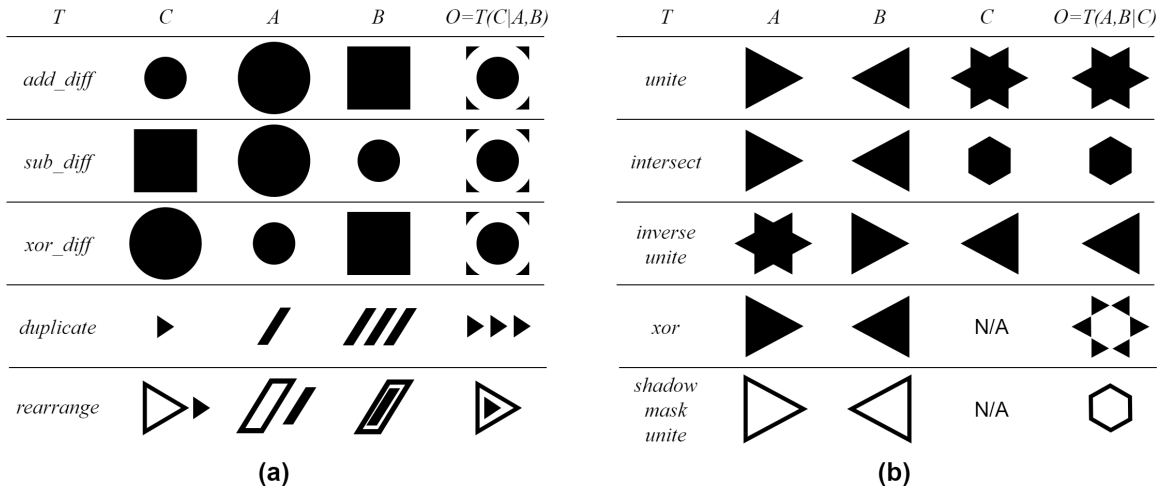

Figure 4. Illustrations of set transformations used in our models: (a) Given an analogy *A:B::C:?* and an unary set transformation T, the output image is $O = T(C|A, B)$, where C is the input, and B and C are parameters of T. (b) Given an analogy *A:B:C::D:E:?* and a binary set transformation T, the output image is $O = T(A, B|C)$ when T is applied on *A:B:C*, where A and B are the inputs, and C is a parameter of T, or $O = T(D, E|O')$ when T is applied on *D:E:?*, where O' is any option of the RPM problem.

Table 1. Details of unary, binary, and hybrid unary/binary transformations.

$add\_diff(C\|A, B)$	Calculate $S_A(A, B) = (\cdots, pos_{DA}, D)$. Align D and C using $pos_{DA\|A=C}$. Output $O = C \cup D$.
$sub\_diff(C\|A, B)$	Calculate $S_A(B, A) = (\cdots, pos_{BA}, pos_{DB}, D)$. Align C and D using $pos_{BA\|A=C}$ and pos_{DB}. Output $O = C - D$.
$xor\_diff(C\|A, B)$	Calculate $S(A, B) = (\cdots, pos_{AB})$. Align A and B using pos_{AB}, and calculate $D = A \oplus B$. Align C and D using $pos_{AB\|A=C}$. Output $O = C \oplus D$.
$duplicate(C\|A, B)$	Let O be an empty image of the same size as B. Calculate $S_A(A, B) = (\cdots, pos_{AB}, \cdots)$ and $B = B - A$ aligned by pos_{AB}, and copy C to the position of $pos_{AB\|A=C}$ in O. Repeat this until nothing is left in B. Output O.
$rearrange(C\|A, B)$	Let O be an empty image of the same size as B. Decompose C, A and B into connected components $C_1, C_2, \cdots, C_l$, $A_1, A_2, \cdots, A_m$ and $C_1, C_2, \cdots, C_n$. If $l = m = n$ is false, output a value indicating failure. Otherwise, find a 1-to-1 mapping $f : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, n\}$ that maximizes $\sum_{i=1}^{n} J(A_i, B_{f(i)})$ by calculating $S(A_i, B_j) = (J(A_i, B_j), pos_{A_i B_j})$ for each i and each j. Find another 1-to-1 mapping $g : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, n\}$ that minimizes $\sum_{i=1}^{n} distance(C_i, A_{g(i)})$. Generate O by copying C_i to position of $pos_{A_{g(i)} B_{f(g(i))}\|A_{g(i)}=C_i}$ in O for all i.
$unite(A, B\|C)$	Calculate $S_A(A, C) = (\cdots, pos_{AC}, \cdots)$ and $S_A(B, C) = (\cdots, pos_{BC}, \cdots)$. Align A and B with pos_{AC} and poc_{BC}. Output $O = A \cup B$.
$intersect(A, B\|C)$	Calculate $S_A(C, A) = (\cdots, pos_{CA}, \cdots)$ and $S_A(C, B) = (\cdots, pos_{CB}, \cdots)$. Align A and B with pos_{CA} and poc_{CB}. Output $O = A \cap B$.
$inverse\_unite(A, B\|C)$	Calculate $S_A(B, A) = (\cdots, pos_{BA}, \cdots)$ and $S_A(C, A) = (\cdots, pos_{CA}, \cdots)$. Align A, B and C using pos_{BA} and pos_{CA}. Output image $O = A - (B - C)$.
$xor(A, B)$	Calculate $S(A, B) = (\cdots, pos_{AB})$. Align A and B by pos_{AB}. Output $O = A \oplus B$.
$shadow\_mask\_unite(A, B)$	Let X and Y be the shadows of A and B, where "shadow" is defined to be a copy of an image where any white area surrounded by black in the original image is colored black. Calculate $S(X, Y) = (\cdots, pos_{XY})$. Align X and Y using pos_{XY}, and calculate $M = X \cap Y$. Align A and B using $pos_{XY\|X=A,Y=B}$. Output $O = M \cap (A \cup B)$.
$preserving\_sub\_diff(D, E\|A, B, C)$	Given analogy *A:B:C::D:E:?*, $preserving\_sub\_diff$ works as $sub\_diff(E\|B, C)$. But it requires that $A \subset (B-C)$ and $D \subset (E-O)$, where O is an option. Otherwise, output a value indicating failure. (This transformation NOT shown in Figure 4.)

transformations are based on fundamental affine transformations or set operations, or combinations of these. These transformations extend the original collections proposed in earlier ASTI research (Kunda et al., 2013; Kunda, 2013).

We have nine unary affine transformations in our models: eight rectilinear rotations/reflections, as shown in Figure 3, plus a ninth scaling transformation that doubles the area of the input image.

We have eleven additional set transformations in our models: five unary and five binary, as shown in Figure 4.a and b respectively, plus one additional hybrid unary/binary transformation. Details of each transformation are given in Table 1. Unary transformations are defined relative to analogies between pairs of images, i.e. *A:B::C:?* for images A, B, C. Binary transformations are defined relative to analogies between triplets of images, i.e., *A:B:C::D:E:?* for images A, B, C, D, E. How such analogies are defined within a given RPM problem is described in the next subsection.

2.4 Analogies

An analogy within an RPM matrix reasoning problem is composed of abstract parallel relations between matrix entries. We assume that all the abstract parallel relations in an analogy should be instantiated by a single transformation. Note that while this assumption seems adequate for solving most problems on the Standard Raven's test, items on the Advanced test or other geometric analogy tests may require considering multiple transformations (Carpenter et al., 1990; Kunda, 2015). Figure 5 shows some examples of analogies within an RPM matrix, where the images and the missing part are represented by letters and a question mark. The analogies in our models are composed of either pairs of matrix entries (Figure 5.a and b) or 3-tuples of matrix entries (Figure 5.c and d).

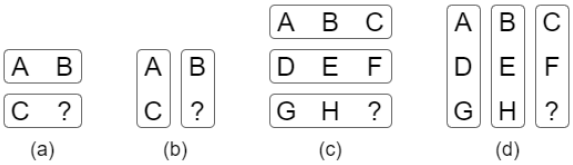

Figure 5. Illustrations of simple analogies in 2×2 and 3×3 RPM problems: given a 2×2 matrix, (a) and (b) show a row analogy *A:B::C:?* and a column analogy *A:C::B:?*; similarly, given a 3×3 matrix, (c) and (d) show row analogies *A:B:C::G:H:?* and *D:E:F::G:H:?*, and column analogies *A:D:G::C:F:?* and *B:E:H::C:F:?*.

In addition to these "simple analogies" in Figure 5, our ASTI+ models also expand these possible analogies drawn from the problem matrix in two important ways. First, for 3×3 matrices, the models further consider several sub-problems, as shown in in Figure 6. For example, consider the simple analogies in Figure 5.c, *A:B:C::G:H:?* and *D:E:F::G:H:?*. They use only two of three rows of the matrix. Thus, we put them in a recursive format, *A:B:C::D:E:F:::D:E:F::G:H:?* as shown in Figure 6.a, where all the rows are taken into account. In this recursive analogy, two sub-problems are created — the first sub-problem is *A:B:C::D:E:?* with *F* as the only option; the second sub-problem is *D:E:F::G:H:?* with the options from the original RPM problem. All the sub-problems should be solved equally well by the correct transformation.

Second, the models capture more sophisticated spatial regularities in a matrix by expanding it in a way that the spatial relation between any two entries in the original matrix still holds everywhere in

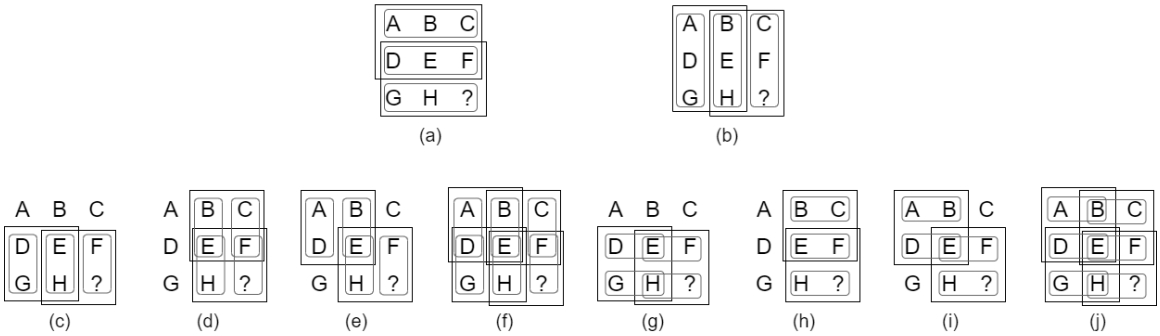

Figure 6. Illustrations of recursive analogies in 3×3 RPM problems: (a) are (b) are 3-tuple analogies and (c) through (j) are pair analogies. Each rectangle or square represents a sub-problem. All the sub-problems in a matrix are equally considered while solving the matrix.

the expanded matrix, and then enclose parts of the expanded matrix with quadrilaterals, as shown in Figure 7. Each quadrilateral contains a matrix that can be used to generate analogies as the original matrix. We follow three reasonable heuristics to enclose these matrices: (1) it should contain all the entries of the original matrix, (2) it should be of the same size as the original matrix, and (3) it should have a *?* at one of the corners. Note that the entries that are grayed out in Figure 7.g are not in any matrix. While we do not necessarily expect that humans use this type of expansion plus quadrilateral strategy, this approach provides a systematic and parsimonious way to capture regularities within a matrix that humans might perceive and reason about, albeit in different ways.

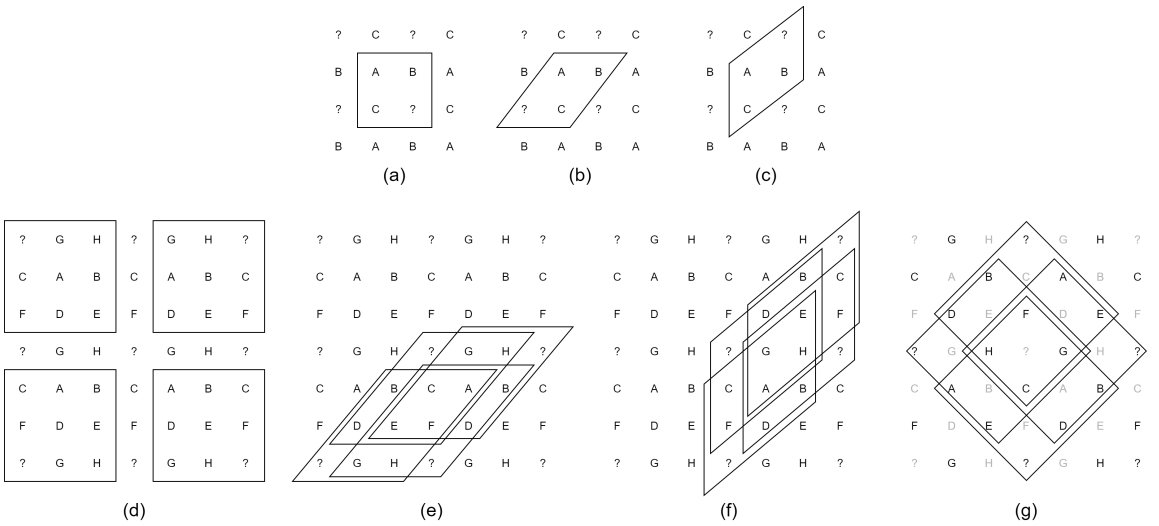

Figure 7. Expanded matrices to generate analogies: (a) through (c) are expanded from the 2×2 matrix in Figure 5, and (d) through (g) are expanded from the 3×3 matrix in Figure 5. Different quadrilaterals are used to enclose matrices to generate analogies.

2.5 General Integration Strategy

All of our ASTI+ models share a 3-stage general strategy for solving an RPM problem. In Stage 1, they try to explain the variations in the matrix with some analogies and transformations. In Stage 2, they verify their explanations by predicting an answer image and comparing it to each option, and/or plugging each option into the matrix and checking if the explanations still hold. In Stage 3, the best explanation, i.e. the best analogy and the best transformation, along with the corresponding answer option, are outputted as the answer.

To quantify "how well" an analogy and a transformation explain the variations across matrix entries, we introduce three kinds of scores based on the Jaccard and the asymmetric Jaccard indices introduced in Section 2.2: (1) the MAT score measures how well an analogy and a transformation explain the variations in the matrix in Stage 1; (2) the O score measures how well an analogy and a transformation explain the variations involving the options in Stage 2; (3) the MATO score, used as the final score to select the answer, is computed from the MAT and O scores.

For example, given the matrix in Figure 5.a, analogy *A:B::C:?* and transformation $flip(X)$: $\mathsf{MAT} = J(flip(A), B)$; $\mathsf{O} = J(flip(C), O)$; and $\mathsf{MATO} = (\mathsf{MAT} + \mathsf{O})/2$. However, score calculation methods are analogy- and transformation-dependent, as described in more details below.

MAT Scores. For transformations in forms of $T(A)$ or $T(A, B)$ (without extra parameters), MAT scores are calculated in the same way as $flip(X)$. For transformations with extra parameters, MAT scores can not be calculated in the same way because we don't know the extra parameters. For example, for $add\_diff(I|S, T)$ and *A:B::C:?*, we can not use $\mathsf{MAT} = J(add\_diff(A|S, T), B)$ because we don't know S and T (but we can use $add\_diff(C|A, B)$ to calculate O score). In this case, the MAT score is calculated as $\mathsf{MAT} = J_A(A, B)$ for $add\_diff(I|S, T)$. Although transformation-specific approaches are taken to calculate MAT scores, it is always a function of one or more Jaccard and asymmetric Jaccard indices of known matrix entries.

O Scores. For transformations using only Jaccard index to calculate MAT scores, the Jaccard index is also used to calculate O scores. For transformations using asymmetric Jaccard index, for example add_diff and sub_diff, asymmetric Jaccard index is always higher than Jaccard index given the same input (see Equation 1 and 2). As a result, transformations measured by asymmetric Jaccard index tend to have higher scores even if their explanations are pretty bad. To regulate such transformations, we calculate multiple Jaccard and asymmetric Jaccard indices, each of which characterizes a distinct aspect of the transformation, and average them to get an O score.

For example, for $add\_diff(C|A, B)$ and *A:B::C:?*, three aspects of the transformation are considered: (1) how much C is a subset of O, where O is an option, (2) how the difference between A and B compares to the difference between C and O and (3) how similar the predicted image is to O. This leads us to $\mathsf{O} = (J_A(C, O) + J(D, D') + J(add\_diff(C|A, B), O)))/3$, where $D = B - A$ and $D' = O - C$ after A, B, C and O are properly aligned.

MATO Scores. MATO scores are weighted averages of MAT and O scores, where the weight is proportional to the number of variations that the score measures. For recursive analogies in 3×3 matrices, scores of the original problem are derived from the scores of sub-problems. Suppose that there are n sub-problems in a recursive analogy. Let MAT_k and O_k be the MAT score and the O score of the k-th sub-problem. The final MAT score is $\mathsf{MAT} = [\sum_{k=1}^{n-1}(\mathsf{MAT}_k + \mathsf{O}_k) + \mathsf{MAT}_n]/(2n-1)$, and the final MATO score is $\mathsf{MATO} = [\sum_{k=1}^{n}(\mathsf{MAT}_k + \mathsf{O}_k)]/2n$.

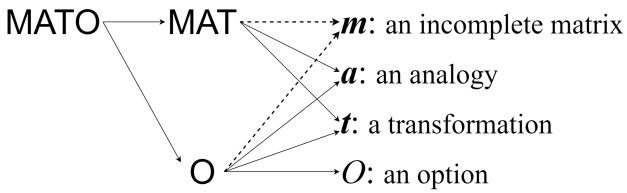

Figure 8. The dependencies of scores: The dashed lines denote partial dependence. Given the relations in an analogy, MAT relies on the entries that are not related to the missing entries while O relies on the entries that are related to the missing entries.

2.6 Specific Integration Strategies: When and What to Maximize

Given the dependencies of scores in Figure 8, the general strategy boils down to an optimization problem in which MATO score is maximized over the analogy $\boldsymbol{a}$, the transformation $\boldsymbol{t}$ and the option O (the matrix $\boldsymbol{m}$ is fixed for an RPM problem). A special heuristic in this optimization is that, often, maximization of MAT score implies the maximization of O score and thus the maximization of MATO score. In other words, the best explanation for the known matrix entries should often still be the best if you plug the options into the matrix. However, this heuristic may not work if there is ambiguity or noise in the problem, or if the reasoning agent does not have sufficient/appropriate transformations for that particular problem.

Therefore, we introduce three specific integration strategies, i.e., each in the form of a specific maximizing process—from totally relying on the heuristic to totally ignoring the heuristic. In particular, given an RPM matrix $\boldsymbol{m}$, an analogy $\boldsymbol{a}$, a transformation $\boldsymbol{t}$ and an option O, MAT score is a function $\mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t})$, O score is a function $\mathsf{O}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}, O)$, and MATO is a function $\mathsf{MATO}(\mathsf{MAT}, \mathsf{O})$. The 3 processes can be written as the optimization problems I, II and III, where I totally relies on the heuristic, III totally ignores the heuristic, and II lies in between.

Models using these three optimization formulations are labeled as following **M-confident** (I), **M-neutral** (II), and **M-prudent** (III) strategies, respectively.

$$\begin{aligned} \mathsf{MATO}^* &= \max_{O} \mathsf{MATO}(\mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}^*, \boldsymbol{t}^*), \mathsf{O}(\boldsymbol{m}, \boldsymbol{a}^*, \boldsymbol{t}^*, O)) \\ \boldsymbol{a}^*, \boldsymbol{t}^* &= \arg\max_{\boldsymbol{a}, \boldsymbol{t}} \mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}) \end{aligned} \tag{I}$$

$$\begin{aligned} \mathsf{MATO}^* &= \max_{\boldsymbol{a}, O} \mathsf{MATO}(\mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}^*), \mathsf{O}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}^*, O)) \\ \boldsymbol{t}^* &= \arg\max_{\boldsymbol{t}} \mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}) \end{aligned} \tag{II}$$

$$\mathsf{MATO}^* = \max_{\boldsymbol{a}, \boldsymbol{t}, O} \mathsf{MATO}(\mathsf{MAT}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}), \mathsf{O}(\boldsymbol{m}, \boldsymbol{a}, \boldsymbol{t}, O)) \tag{III}$$

Since the O score also depends on the option O in Figure 8, it can also serve as the objective function to select the answer from the options. Therefore, we have another 3 optimizations maximizing O, which we label as corresponding O-strategies: **O-confident**, **O-neutral** (IV) ,and **O-prudent** (V). (Note that since MATO is simply a weighted average of MAT and O, O-confident is equivalent to M-confident (I)).

$$\begin{aligned} \mathsf{O}^* &= \max_{\boldsymbol{a},O} \mathsf{O}(\boldsymbol{m},\boldsymbol{a},\boldsymbol{t}^*,O) \\ \boldsymbol{t}^* &= \arg\max_{\boldsymbol{t}} \mathsf{MAT}(\boldsymbol{m},\boldsymbol{a},\boldsymbol{t}) \end{aligned} \tag{IV}$$

$$\mathsf{O}^* = \max_{\boldsymbol{a},\boldsymbol{t},O} \mathsf{O}(\boldsymbol{m},\boldsymbol{a},\boldsymbol{t},O) \tag{V}$$

A natural way to consider these optimization problems is that they correspond to the reasoning agents from being extremely confident to being extremely prudent. For example, a very confident agent (I) might come up with only one explanation (i.e. an analogy and a transformation) for what is observed in the matrix, and then verify this explanation against the options, while a very prudent agent (Optimization III and V) enumerates all the possibilities.

3. Experimental Results

First, we compare integration strategies, given the complete set of transformations and analogies. Models were run against all 60 scanned problems from the Standard RPM test.Results are shown in Figure 9. The M-neutral strategy always ties with the M-prudent strategy, solving 57/60 problems. The M-confident strategy performs slightly worse, solving 55/60 problems.

Interestingly, while the confident strategy performs worst in maximizing MATO in Figure 9.a, it performs best in maximizing O in Figure 9.b. Meanwhile, O-neutral and O-prudent strategies in Figure 9.b contrast sharply with their counterparts in Figure 9.a. In particular, the more the strategy relies on the heuristic that we mentioned in Section 2.5, the more the performance degrades from maximizing MATO to maximizing O. We surmise that this is because the RPM is designed to have distractors of very high O and very low MAT. In other words, these distractors work like traps for strategies that maximize only O scores, which is consistent with observations that people often make errors of "repetition" while solving RPM problems (Kunda et al., 2016).

The bubble tables in Figure 9.c and 9.d show the details of each problem's answers given by different strategies. MAT, O and correctness of the answers are encoded as size, darkness and hue of bubbles (correct answers are in blue and incorrect ones are in red). Note that the "signed" O score is only for visualization to distinguish between correct and incorrect answers, and the real scores in our models are always positive in $[0, 1]$. Figure 9.c also shows a subtle difference between strategies: although different strategies can have the same answer to a problem, they may come to the same answer via different analogies and transformations; otherwise blue bubbles in any single column should be of the same size and color.

If we put all the bubbles in a 2-D plane of MAT and O scores, we will have the scatterplots in Figure 9.e and 9.f. Since most of the points in Figure 9.e represent problems that are correctly solved, and these points are mostly located near or below the diagonal, we hypothesize that, for a "naive" (with little prior knowledge about RPM) participant or computational model to solve an RPM problem, a good explanation for the known matrix entries matters more than a good explanation of how an option can be matched. Recall that MAT and O are measurements of these two explanations. On the flip side, many more points, representing incorrect answers, are located above

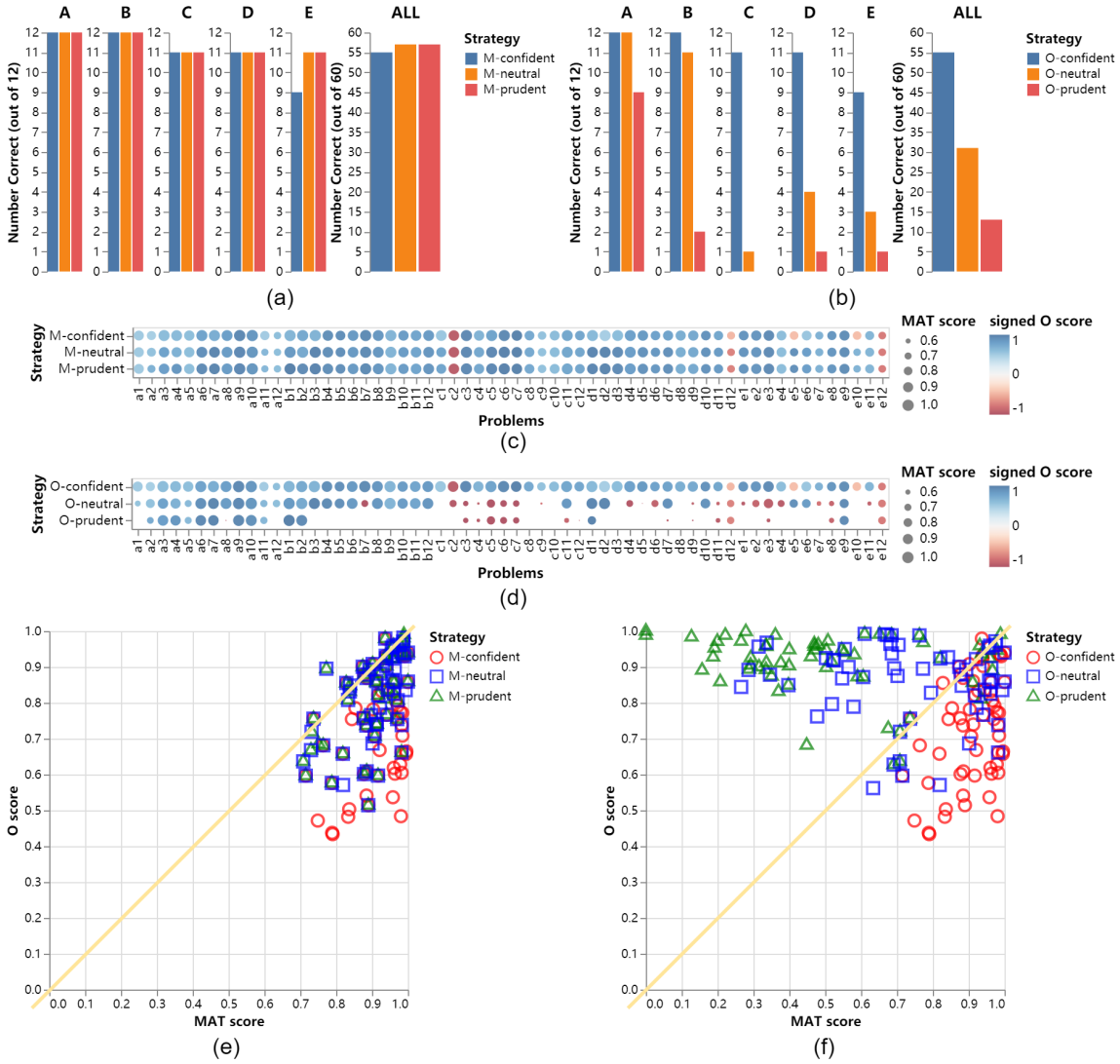

Figure 9. Comparison between the strategies given the complete set of analogies and transformations: (a) and (b) show numbers of problems correctly solved by each strategy in every single problem set and the entire standard RPM test; (c) and (d) are bubble tables showing MAT and O scores for each strategy and each problem, where red bubbles are incorrect answers and blue bubbles are correct answers; (e) and (f) are scatterplots where each bubble in (c) and (d) is a datapoint in the 2-D plance of MAT and O. Note that (a), (c) and (e) are for strategies maximizing MATO while (b), (d) and (f) are for strategies maximizing O.

the diagonal in Figure 9.f, which further consolidates this hypothesis. This hypothesis based on MAT and O scores is consistent with previous suggestions that high-achieving testees take the options less into account, or none at all except comparing their predictions with the options, than low-achieving testees (Bethell-Fox et al., 1984; Carpenter et al., 1990; Lovett & Forbus, 2017).

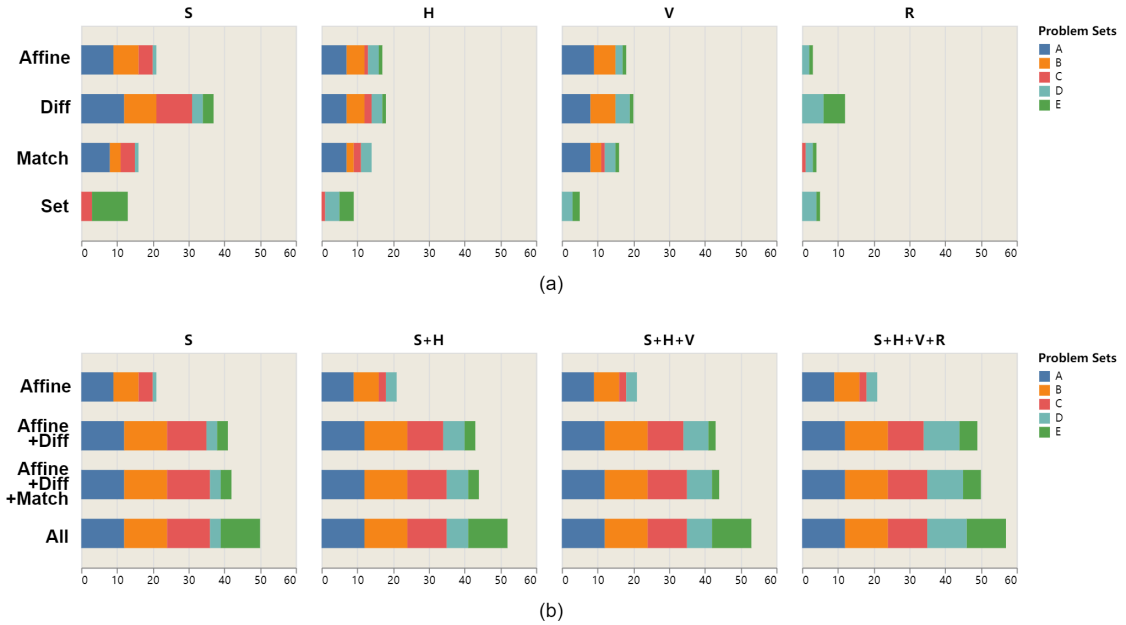

Figure 10. Stack bar charts of numbers of problems correctly solved by M-prudent strategy using different analogy groups and transformation groups.

Next, we compare model configurations all using the same M-prudent strategy but with different subsets of transformations and analogies. We divide transformations into 4 groups: Affine={all the affine transformations}, Diff={add_diff, sub_diff, xor_diff}, Match={duplicate, rearrange} and Set={unite, intersect, inverse_unite, preserving_sub_diff, xor, shadow_mask_unite}. We also divide analogies into 4 groups according to shapes of quadrilaterals: Group S from Figure 7.a and 7.d, Group H from Figure 7.b and 7.e, Group V from Figure 7.c and 7.f, and Group R from Figure 7.g. Figure 10 shows the performance of different combinations of analogy and transformation groups using the M-prudent strategy. The groups are arranged in a 1-to-1 way in Figure 10.a and in a progressive way in Figure 10.b. The strength and the weakness of each analogy group and each transformation group clearly emerge from Figure 10.a. For example, S analogies plus Diff transformations are really good at problems in Set A, B and C, whereas R analogies and Set transformations specialize in solving Set D and E but work pretty bad on Set A, B, and C. In Figure 10.b, increases can be seen in both vertical and horizontal directions. However, the vertical increases are more significant than the horizontal increases. This does not mean that transformations are more important than analogies, because, as we can see in Figure 10.a, the S group outperforms H, V and R for every transformation group, and most of the problems in Set A, B and C solved by H, V and R can also be also solved by S with a different transformation. We might expect to see more variation across analogy groups if they are defined at a finer level.

4. Related Work

4.1 Knowledge-Based Models

Knowledge-Based Models using Propositional Representations. While not explicitly an RPM model, Evans' early ANALOGY program (Evans, 1964) solved geometric analogy problems in the form of *A:B::C:{five options}* by applying a predefined set of spatial transformations to account for variation in the analogy source. Carpenter et al. (1990) similarly used 5 predefined rules to explain the variations in Advanced RPM matrices, and conducted experiments around working memory and subgoaling. Some recent works brought propositional representations to a finer level of components of geometric objects, rather than the entire geometric objects. One of the most well-known works is CogSketch plus Structure-Mapping Engine (Falkenhainer et al., 1989; Lovett et al., 2010; Forbus et al., 2011; Lovett & Forbus, 2017), where CogSketch encodes qualitative spatial relations between 2-D objects, and Structure-Mapping Engine compares the relations using structure-mapping theory (Gentner, 1983). Prade and Richard (2009; 2010; 2013) also provide great theoretical and systematical interpretations of analogies and apply them on various analogical reasoning tasks such as progressive matrices. Another distinctive work is the anthropomorphic method (Cirillo & Ström, 2010; Strannegård et al., 2013), which solves the RPM problems without looking at the options. For each problem, it maintains a hierarchical representation of the answer and keeps updating it with analogies and transformations. Different from all the above models is a work from a computational neuroscience perspective by Rasmussen and Eliasmith (2011), where each transformation is implemented through spike neuron models using Vector Symbolic Architecture (Gayler, 2004) and Neural Engineering Framework (Eliasmith & Anderson, 2004).

Knoweldge-Based Models using Visual Representations. Kunda, McGreggor and Goel (2009; 2010; 2011; 2012; 2013; 2014; 2014) proposed several computational models that work directly on the visual representations of geometric objects, including the original ASTI model extended here as well as the fractal model. Shegheva and Goel (2018) proposed another interesting model, which represents rules of variations with graphical models using Markov Random Fields.

4.2 Data-Driven Models

Data-Driven Models using Visual Representations. Recent approaches from machine learning have tackled RPM-like problems using mostly neural-network-based techniques. The general machine-learning solution for RPM problems is: (1) each answer option is plugged into the matrix, (2) sequences of images of rows, columns or the entire matrix are fed into a feature-extraction module, (3) scores of how well the option fits into the row, the column or the matrix are computed on the features, and (4) the answer is selected according to the scores. Two Raven-like datasets, Procedurally Generated Matrices (PGM) (Santoro et al., 2018) and Relational and Analogical Visual rEasoNing (RAVEN) (Zhang et al., 2019), have been proposed and widely used to train machine learning models.

Zhuo and Kankanhalli (2020) show that an RPM problem can be solved by both supervised learning and unsupervised learning. For supervised learning, they experiment with ResNet models pre-trained on ImageNet, which, nonetheless consists of very distinct images from RPM problems, can still perform quite well on RAVEN dataset. For unsupervised learning, although no scores

of the options are specified for training, they introduced pseudo-labels (scores) to train the models: they plug each answer option into a 3×3 matrix to get 8 different last rows; the first 2 rows and the 8 last rows are assigned pseudo-labels of $[1, 1, 0, 0, 0, 0, 0, 0, 0, 0]$, which allows a standard CNN+dropout+fully-connected model to be trained in a supervised way.

Santoro et al. (2017) proposed an interesting relational network (RN) module which explicitly computes the pairwise relation (as a vector) between any two input objects. It is a plug-and-play module for many existing deep learning models, which originally work on a set of input objects, to work on the Cartesian product of the set of input objects. Based on RN modules, WReN model and its extensions (Santoro et al., 2018; Jahrens & Martinetz, 2020; Steenbrugge et al., 2018; van Steenkiste et al., 2019) are proposed and tested against the PGM dataset.

Different from the general machine-learning solution, Hua and Kunda (2019) proposed the first generative data-driven model for solving RPM problems (to the best of our knowledge), which utilizes GAN networks to generate the predicted images for missing entries, and the answers are selected through image similarities between the predicted images and the options.

Using Both Visual and Propositional Representations. The RAVEN dataset is different from PGM dataset in that it includes both visual and propositional descriptions of geometric objects. To incorporate propositional descriptions into machine learning models, Zhang et al. (2019) proposed the Dynamic Residual Tree (DRT) module to add the propositional structural information of geometric objects into the image features. Ahmed et al. (2019) proposed another model that used two LSTMs for reasoning through propositional and visual representations respectively.

5. Conclusion and Future Work

In this paper, we have described a new search hierarchy framework for describing reasoning on Raven's matrix problems, including model variations in transformations, analogies, and integration strategies. We show that a specific configuration of the ASTI+ model can solve 57/60 problems on the Raven's Standard Progressive Matrices test, using scanned images from the paper test booklet as inputs. We further demonstrate that test performance can vary widely not only as a function of transformations and analogies that a model might use, but also the higher-level integration strategy, i.e., when and how, across analogies and transformations, the model chooses to perform its maximization calculations.

In tasks such as the RPM where *eductive ability* is required to extract information from a new situation, redundant information of regularity often exists; otherwise, ambiguity can hardly be eliminated because little prior knowledge is known about this situation. Therefore, methods for representing, identifying, and exploiting such redundancies are crucial to solving the task. Generally, analogy is often used for this purpose, as in our ASTI+ models. By varying the configuration of the ASTI+ model, we are actually varying its ability to identify and represent redundancies, and the extent to which it can exploit these redundancies to inform is answer.

The implication of this work in term of intelligence is twofold. First, for artificial intelligence, analogical ability is an unavoidable issue if we expect our agents to be able to operate in new situations. Second, for human intelligence, analogical ability contributes significantly to individual differences in eductive ability. The ASTI+ models demonstrate analogical reasoning framed as

exhaustive searches through various pre-defined analogical spaces. Humans' analogical reasoning is far more sophisticated than this kind of explicit search through a predefined analogy space; instead, analogy spaces must be inferred, and every analogy is built "on demand." This raises an interesting and central question for AI agents, i.e., how agents can develop or bootstrap analogy spaces and select from them as humans do, instead of resorting to explicit searches over predefined search spaces. Additional important open areas for future work are discussed below.

We used some parameters in our models that were fit specifically to the specific test problem inputs used in our experiments—for example, the threshold to convert grayscale images to binary images, and the threshold to filter out noise pixels. The transformations and similarity metrics used by the model are, in turn, also sensitive to these parameters. To remove these dependencies, we can either integrate advanced image processing techniques to improve the input image representation, or develop more robust transformations and similarity metrics.

Our current models use only a single analogy and a single transformation in an explanation. However, considering multiple analogies and transformations simultaneously is likely important, especially for more advanced problems beyond the standard Raven's test (Carpenter et al., 1990; Kunda, 2015). Then, how to coordinate multiple pathways and merge the results from these pathways becomes an important open question.

Going one step further, virtually all extant computational RPM models, this one included, essentially use a single reasoning pipeline to solve every problem. However, there is ample evidence from psychology and neuroscience that many people use multiple strategies to solve Raven's problems, including within the context of a single testing session. For example, studies have found behavioral (DeShon et al., 1995) and neural (Prabhakaran et al., 1997) differences across test items linked to visual versus verbal problem solving strategies (see Kunda et al. (2013) for a review of these visual/verbal strategy differences); many other dimensions of strategy differences likely exist as well. How do people manage multiple strategies, and meta-cognitively perform strategy selection and switching for appropriate problems? And, correspondingly, how might an intelligent agent benefit from similar strategy flexibility during complex problem solving as on the Raven's test?

Finally, while the definitions of analogies and strategies were performed manually in this research, there is an enormous and looming question of how humans induce such strategies online, which (to our knowledge) no prior AI systems have accomplished on the Raven's test (Hernández-Orallo et al., 2016). (Even RPM models that use learning-based approaches still require the system designer to define, for example, the notion of maximizing a function between entries in the problem matrix and each answer option.) Research in program induction may provide one path to tackle this thorny question (Schmid & Kitzelmann, 2011), including how strategies might be induced in the first place as well as adapted from problem to problem.

Acknowledgements

We would like to thank Ashok Goel for extensive discussions and contributions to earlier phases of this research, as well as the anonymous reviewers for their helpful feedback. This work was supported in part by NSF Award #1730044.

References

Ahmed, S. A., Dogra, D. P., Kar, S., Roy, P. P., & Prasad, D. K. (2019). Can we automate diagrammatic reasoning? *Pattern Recognition*.

Bethell-Fox, C. E., Lohman, D. F., & Snow, R. E. (1984). Adaptive reasoning: Componential and eye movement analysis of geometric analogy performance. *Intelligence*, *8*, 205–238.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. *Psychological review*, *97*, 404.

Cirillo, S., & Ström, V. (2010). *An anthropomorphic solver for raven's progressive matrices*. Master's thesis.

DeShon, R. P., Chan, D., & Weissbein, D. A. (1995). Verbal overshadowing effects on raven's advanced progressive matrices: Evidence for multidimensional performance determinants. *Intelligence*, *21*, 135–155.

Eliasmith, C., & Anderson, C. H. (2004). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.

Evans, T. G. (1964). *A program for the solution of a class of geometric-analogy intelligence-test questions.*. Technical report, AIR FORCE CAMBRIDGE RESEARCH LABS LG HANSCOM FIELD MASS.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, *41*, 1–63.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2011). Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, *3*, 648–666.

Gayler, R. W. (2004). Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, *7*, 155–170.

Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., & Dowe, D. L. (2016). Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, *230*, 74–107.

Hespos, S. J., Anderson, E., & Gentner, D. (2020). Structure-mapping processes enable infants' learning across domains including language. In *Language and concept acquisition from infancy through childhood*, 79–104. Springer.

Hua, T., & Kunda, M. (2019). Modeling gestalt visual reasoning on the raven's progressive matrices intelligence test using generative image inpainting techniques. *arXiv preprint arXiv:1911.07736*.

Jahrens, M., & Martinetz, T. (2020). Solving raven's progressive matrices with multi-layer relation networks. *arXiv preprint arXiv:2003.11608*.

Kunda, M. (2013). *Visual problem solving in autism, psychometrics, and ai: the case of the raven's progressive matrices intelligence test*. Doctoral dissertation, Georgia Institute of Technology.

Kunda, M. (2015). Computational mental imagery, and visual mechanisms for maintaining a goal-subgoal hierarchy. *Proceedings of the Third Annual Conference on Advances in Cognitive Systems (ACS)* (p. 4).

Kunda, M., McGreggor, K., & Goel, A. (2009). Addressing the raven's progressive matrices test of "general" intelligence. *2009 AAAI Fall Symposium Series*.

Kunda, M., McGreggor, K., & Goel, A. (2010). Taking a look (literally!) at the raven's intelligence test: Two visual solution strategies. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Kunda, M., McGreggor, K., & Goel, A. (2011). Two visual strategies for solving the raven's progressive matrices intelligence test. *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Kunda, M., McGreggor, K., & Goel, A. (2012). Reasoning on the raven's advanced progressive matrices test with iconic visual representations. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the raven's progressive matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, *22*, 47–66.

Kunda, M., Soulières, I., Rozga, A., & Goel, A. K. (2016). Error patterns on the raven's standard progressive matrices test. *Intelligence*, *59*, 181–198.

Little, D. R., Lewandowsky, S., & Griffiths, T. L. (2012). A bayesian model of rule induction in raven's progressive matrices. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Lovett, A., & Forbus, K. (2017). Modeling visual problem solving as analogical reasoning. *Psychological review*, *124*, 60.

Lovett, A., Forbus, K., & Usher, J. (2010). A structure-mapping model of raven's progressive matrices. *Proceedings of the Annual Meeting of the Cognitive Science Society*.

McGreggor, K., & Goel, A. (2014). Confident reasoning on raven's progressive matrices tests. *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

McGreggor, K., Kunda, M., & Goel, A. (2014). Fractals and ravens. *Artificial Intelligence*, *215*, 1–23.

Memisevic, R., & Hinton, G. E. (2010). Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural computation*, *22*, 1473–1492.

Michelson, J., Palmer, J. H., Dasari, A., & Kunda, M. (2019). Learning spatially structured image transformations using planar neural networks. *arXiv preprint 1912.01553*.

Prabhakaran, V., Smith, J. A., Desmond, J. E., Glover, G. H., & Gabrieli, J. D. (1997). Neural substrates of fluid reasoning: an fmri study of neocortical activation during performance of the raven's progressive matrices test. *Cognitive psychology*, *33*, 43–63.

Prade, H., & Richard, G. (2009). Analogy, paralogy and reverse analogy: Postulates and inferences. *Annual Conference on Artificial Intelligence* (pp. 306–314). Springer.

Prade, H., & Richard, G. (2010). Reasoning with logical proportions. *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*.

Prade, H., & Richard, G. (2013). From analogical proportion to logical proportions. *Logica Universalis*, *7*, 441–505.

Rasmussen, D., & Eliasmith, C. (2011). A neural model of rule generation in inductive reasoning. *Topics in Cognitive Science*, *3*, 140–153.

Raven, J., Raven, J. C., & Court, J. H. (1998). Manual for raven's progressive matrices and vocabulary scales.

Santoro, A., Hill, F., Barrett, D., Morcos, A., & Lillicrap, T. (2018). Measuring abstract reasoning in neural networks. *International Conference on Machine Learning* (pp. 4477–4486).

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. *Advances in neural information processing systems* (pp. 4967–4976).

Schmid, U., & Kitzelmann, E. (2011). Inductive rule learning on the knowledge level. *Cognitive Systems Research*, *12*, 237–248.

Shegheva, S., & Goel, A. (2018). The structural affinity method for solving the raven's progressive matrices test for intelligence. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Steenbrugge, X., Leroux, S., Verbelen, T., & Dhoedt, B. (2018). Improving generalization for abstract reasoning tasks using disentangled feature representations. *arXiv preprint arXiv:1811.04784*.

van Steenkiste, S., Locatello, F., Schmidhuber, J., & Bachem, O. (2019). Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems* (pp. 14222–14235).

Strannegård, C., Cirillo, S., & Ström, V. (2013). An anthropomorphic method for progressive matrix problems. *Cognitive Systems Research*, *22*, 35–46.

Zhang, C., Gao, F., Jia, B., Zhu, Y., & Zhu, S.-C. (2019). Raven: A dataset for relational and analogical visual reasoning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5317–5327).

Zhuo, T., & Kankanhalli, M. (2020). Solving raven's progressive matrices with neural networks. *arXiv preprint arXiv:2002.01646*.