Evaluating Understanding on Conceptual Abstraction Benchmarks

Victor Vikram Odouard[1], Melanie Mitchell[1]

[1]Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501 USA

Abstract

A long-held objective in AI is to build systems that understand concepts in a humanlike way. Setting aside the difficulty of building such a system, even trying to evaluate one is a challenge, due to present-day AI's relative opacity and its proclivity for finding shortcut solutions. This is exacerbated by humans' tendency to anthropomorphize, assuming that a system that can recognize one instance of a concept must also understand other instances, as a human would. In this paper, we argue that understanding a concept requires the ability to use it in varied contexts. Accordingly, we propose systematic evaluations centered around concepts, by probing a system's ability to use a given concept in many different instantiations. We present case studies of such an evaluations on two domains—RAVEN (inspired by Raven's Progressive Matrices) and the Abstraction and Reasoning Corpus (ARC)—that have been used to develop and assess abstraction abilities in AI systems. Our *concept-based* approach to evaluation reveals information about AI systems that conventional test sets would have left hidden.

Keywords

abstraction, analogy, concepts, machine learning, evaluation

1. Introduction

What unites chain-link fences, high prices, entrance exams, and import tariffs? They are all different kinds of *barriers*. Your understanding of physical barriers may have helped you quickly intuit how chess pieces move (and the fundamental difference between the knight and the other pieces) from very few examples. It may have helped you relate to a friend struggling with credit card debt, even when your obstacles are very different. It may have helped you describe how being jet-lagged sometimes feels like "hitting a wall." These examples illustrate the importance of abstract concepts in few-shot learning, generalization, emotional intelligence, and communication. Such examples display the intuition behind Barsalou's definition of a concept: "a competence or disposition for generating infinite conceptualizations of a category" [1]. In short, understanding the world entails being able to recognize and generate concepts in both concrete and abstract forms.

Early pioneers suggested that their AI summer project might lead to blueprints for machines that could "form abstractions and concepts" [2]. More than six decades later, AI systems are still extremely limited in this regard: they have yet to surmount the "barrier" of understanding [3].

Evaluating a system's understanding of concepts and abstractions is challenging. AI systems are known to be susceptible to shortcut learning, such as recognizing pictures of animals by looking for blurry backgrounds [4] or pictures of cows by looking at surrounding landscapes [5]. More insidiously, certain image classifiers can be fooled into classifying, say, school buses as ostriches by changing the picture in ways indiscernible to human viewers [6].

In this paper, we propose systematic assessments centered around concepts—a *concept-based* approach—to evaluate understanding in AI systems. This approach involves (1) identifying a set of concepts a system should know and (2) designing sets of questions probing for the grasp of these concepts using a variety of instantiations of each concept.

One of the important pillars of the traditional train/test paradigm in machine learning—that the training and test sets be independent and identically distributed (IID)—is violated with our concept-based evaluation method. In order to probe understanding by creating varied concept instantiations, the examples used for evaluation may not be drawn from the same "distribution" as the training set. Furthermore, the examples in evaluation set will likely not be independent in any sense, since they are created by varying specific concepts. In two case studies, we find that our evaluation method reveals important information about a system's ability to understand concepts that might be hidden using a conventional IID test set.

We created concept-based evaluations for two domains that have been used to develop and assess conceptual abstraction abilities in AI systems: RAVEN [7] (inspired by Raven's Progressive Matrices (RPMs) [8]) and the Abstraction and Reasoning Corpus (ARC) [9]. Figure 1 shows a sample problem in the RAVEN domain. Each such problem consists of a three-by-three matrix (Figure 1 left) in which each of 8 matrix components is a

EBeM'22: AI Evaluation Beyond Metrics, July 24, 2022, Vienna, Austria
vo47@cornell.edu (V. V. Odouard); mm@santafe.edu (M. Mitchell)

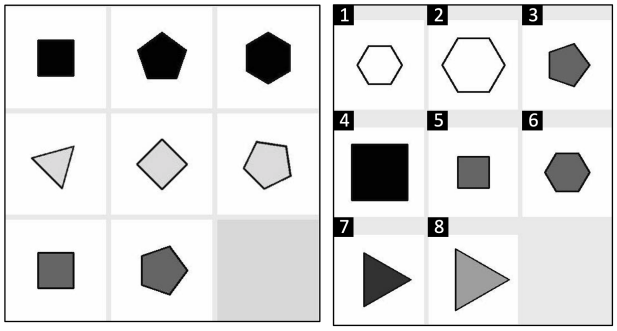


Figure 1: A sample problem in the RAVEN domain. Each row gives polygons with increasing number of sides, with size and color (i.e., gray scale) staying fixed; the correct answer is choice 6.

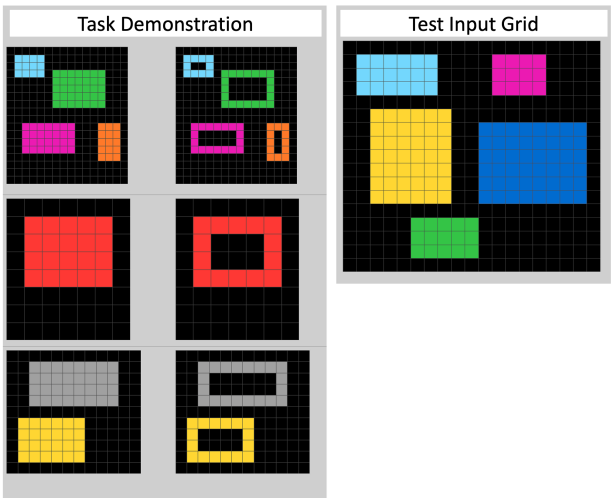


Figure 2: A sample problem ("task") in the ARC domain. The solver's challenge is to generate a grid that transforms the test input grid in the same way as in the task demonstrations. Best viewed in color.

figure involving geometric shapes, with some relationship between the figures in the rows and columns. The ninth component is missing, and the task is to fill in the missing component with one of a set of eight candidate answers (Figure 1 right).

ARC problems (termed "tasks" in [9]) present a number of "demonstration" pairs of grids which are related via a transformation rule, asking the solver to "do the same thing" (i.e., apply the same transformation) to a new "test" input grid. Figure 2 shows a sample task in the ARC domain. The solver's challenge is to generate a new grid that transforms the test input grid analogously to the transformations in the demonstration grids. The concepts used in the ARC domain were inspired by Spelke's proposals for core knowledge systems [10] such as spatio-temporal relations (inside, above, next-to), object attributes (shape, size, color, boundary), transformations (rotate, shift, extend), and more general relations (progression, sameness, part-whole). Notably, ARC tasks require the solver to *generate* an answer, rather than choose among given candidate answers, as in RAVEN, providing the potential for more insight into the understanding of the solver [9].

2. Prior Results on RAVEN

The RAVEN domain was inspired by Raven's Progressive Matrices (RPMs), a kind of IQ test that has been used to measure "fluid intelligence" in humans for many decades [8]. There have been numerous efforts to apply AI and machine learning methods to RPM-like problems (e.g., [7, 11, 12, 13, 14, 15, 16, 17], among many others). Recently many groups have applied deep neural networks (DNNs) to such problems, but given that DNNs need large numbers of training examples, these efforts require meth-

ods for procedural generation of these examples. The creators of the RAVEN dataset [7] developed one such method (another method was used to generate the PGM dataset [11]). To generate a RAVEN problem, the system sampled from a hierarchical stochastic image grammar [7], which offered different possible layouts for the matrix components (e.g., center, inside/outside, grid), and within each layout it offered a choice of shapes (e.g., circle, square, triangle, pentagon) with different attributes to be chosen (e.g., color, size, angle), where each attribute is constrained to be one of a small number of values. The grammar also enforced one of a choice of relationships between matrix elements in a row (e.g., constant, progression, arithmetic); see [7] for details. The authors generated 70,000 problems total, splitting RAVEN into 42,000 training, 14,000 validation, and 14,000 test examples.

In the paper detailing the RAVEN dataset, Zhang et al. [7] reported human performance on RAVEN's test set at 84% accuracy on average. Several subsequent papers reported deep learning methods which surpassed human performance on this dataset (e.g., [18, 19]).

The original RAVEN dataset, however, had a bias in its answer-generation method: answer choices were generated by taking the correct answer and modifying an attribute, allowing solvers to take the majority vote for each attribute to get the correct answer. In fact, networks trained *solely* on the answer choices could attain over 90% accuracy [13]. To remedy this shortcoming, other groups generated modified versions of the answer choices in RAVEN using methods that that seem to be less exploitable. The new versions of RAVEN included RAVEN-FAIR [12] and I-RAVEN [13]. Several groups have since reported test-set accuracies on these new versions that significantly surpass the human performance benchmark of 84% (e.g., [12, 17, 20]).

3. Concept-Based Evaluations for RAVEN

When a program (e.g., a DNN) exhibits high accuracy on the RAVEN dataset, does the program *understand* the concepts expressed in the problems it solved, as a human would? And when a program for solving ARC problems correctly solves a task, to what extent is the program capturing the abstract reasoning abilities the dataset's name implies?

As we have argued above, the way to answer these questions is to evaluate these programs on systematic variations of the *concepts* that they purport to understand. Neither the RAVEN nor ARC datasets (nor any other abstraction datasets that we are aware of) provides this kind of evaluation. In this section we demonstrate how such an evaluation can be carried out on programs that score high on the RAVEN test set.

We first selected two high-performing models from the RAVEN literature: the Multi-scale Relation Network (MRNet, [12]) and the Scattering Compositional Learner (SCL, [17]). For both these systems, the authors made the code publicly available. We then trained both systems on 30,000 RAVEN training examples—ones that used five of the seven layouts available (Center, 2×2Grid, 3×3Grid, Out-InCenter, and Out-InGrid)[1] We then evaluated the trained system on 10,000 RAVEN test examples that used these layouts.[2] The resulting accuracies on these test examples were 73% for MRNet and 89% for SCL.

We then chose two concepts that are present in RAVEN problems: *Sameness* and *Progression*. Both MRNet and SCL were trained on problems involving some version of these concepts, and both were correct on some instances of these concepts in the RAVEN test set. In order to probe the degree to which these systems grasp these two concepts, we manually created new problems that systematically vary these concepts, by instantiating these concepts using different attributes.

In all *Sameness* problems, the relevant relationship in each row is that one or more attributes remain constant. In the RAVEN domain, the possible attributes include shape, size, color (i.e., gray scale), position, row, column, number, angle, and whether one object is inside or outside another object. Figure 3 shows four sample Sameness problems from our evaluation set.

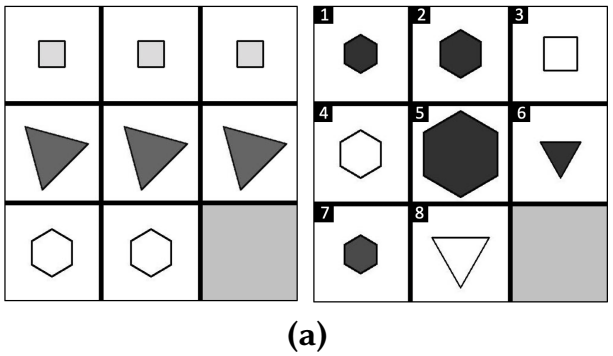
In all *Progression* problems, the relevant relationship in each row is an increase (or decrease) in the value of one or more attributes. Figure 4 shows four sample Progression problems from our evaluation set.

These samples give a flavor of the problem variations we created around each concept. Our evaluation consisted of 210 Sameness and 80 Progression problems, designed to instantiate the concepts in ways that we believe would be relatively easy for humans to understand.[3] The evaluation results are given in Table 1. For both MRNet and SCL, the accuracy on our concept variations are substantially lower than the programs' RAVEN test set accuracy would predict, indicating that their grasp of these general abstract concepts is lacking.

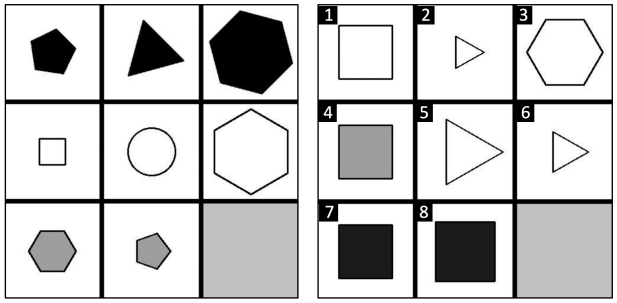
[1] Because these two models scored each answer individually without any comparison between answers, the models were not affected by the answer-generation bias of the original RAVEN dataset we described above. Thus we used the original version to train and evaluate them.

[2] For the sake of time and simplicity, we omitted the Left-Right and Up-Down layouts, which split each matrix component into two.

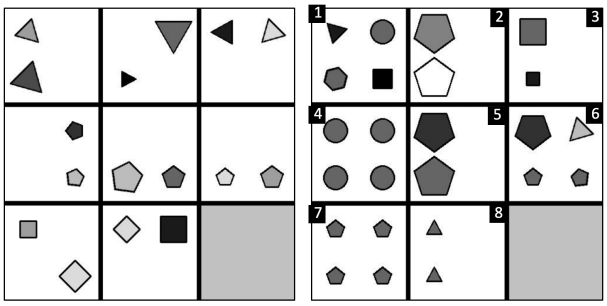
[3] Our Sameness and Progression problems can be downloaded from https://melaniemitchell.me/EBeM2022/RavenVariations.zip.



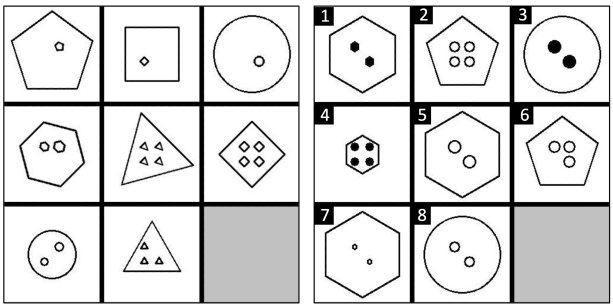
(a)



(b)



(c)



(d)

Figure 3: Four RAVEN variations on the concept *Sameness*. In Problem (a) all attributes remain constant along each row. In Problem (b) color stays constant; in Problem (c) number and shape stay constant, and in Problem (d) in each matrix component, the inner object is the same shape as the outer object. Both SCL and MRNet get the correct answer on (a) and (b), but answer incorrectly on (c) and (d).

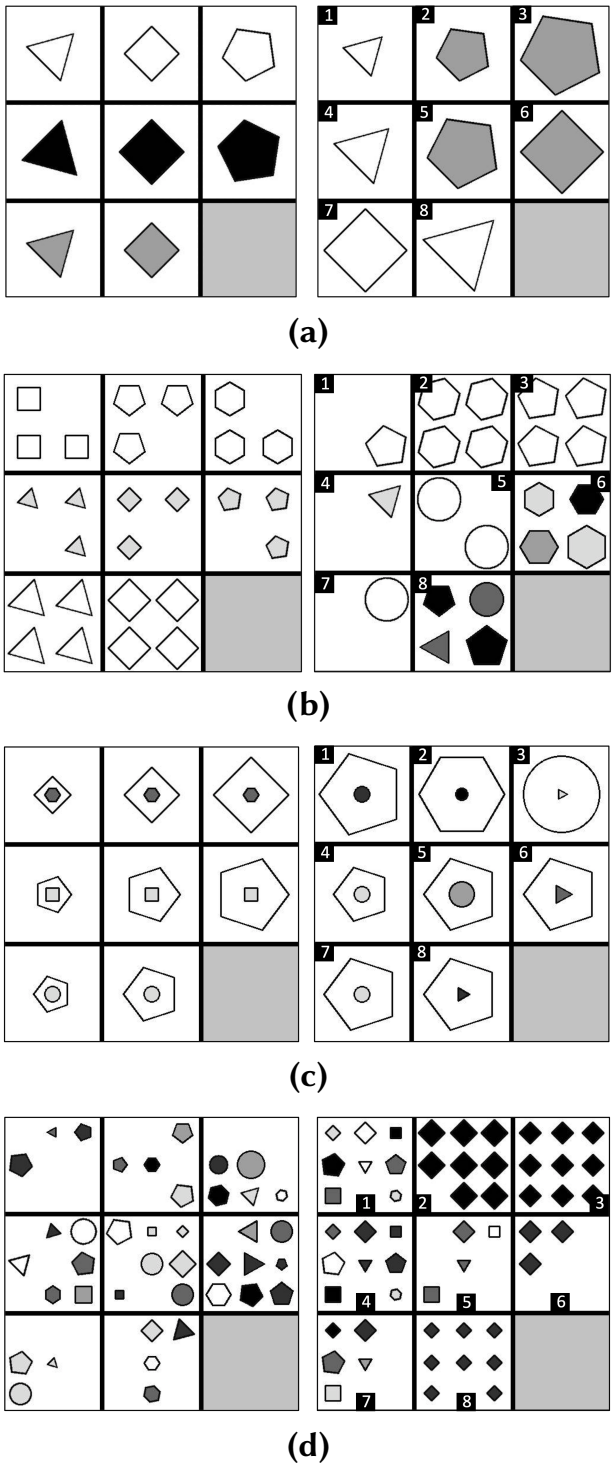


Figure 4: Four RAVEN variations on the concept *Progression*. Problem (a) has a progression in the number of sides of the figure along each row; other attributes stay constant. Problem (b) has the same relationship, but with multiple objects in different positions. In Problem (c) the progression relation is in the size of the outer figure, and in Problem (d) it is in the number of objects. SCL chose the correct answer in all but (d) whereas MRNet was correct on (a) and (c) but not on (b) and (d).

Model	RAVEN Test Set (10,000 problems)	Sameness Variations (210 problems)	Progression Variations (80 problems)
MRNet	73%	49%	44%
SCL	89%	62%	68%

Table 1
Accuracy of MRNet and SCL on original RAVEN test set, and on our concept variations.

4. Prior Results on ARC

Deep learning systems such as MRNet and SCL typically lack transparency. Given their large numbers of parameters and training on large IID datasets, they are susceptible to shortcut learning—that is, learning subtle statistical correlations between their input and the correct answers that don't require actual concept understanding [5]. Such shortcuts are more likely when a system solving problems is allowed to choose from a set of candidate answers, rather than having to generate its own answer. Moreover, the procedural generation of examples—essential for creating sufficiently large training sets—can be susceptible to overt and subtle biases.

Chollet's ARC dataset [9] was created to avoid these pitfalls of deep learning approaches and to be a better method of assessing true abstraction abilities. Unlike RAVEN and related abstraction datasets, ARC focuses on few-shot learning. As shown in Figure 2, each ARC task can be considered a few-shot-learning task: given a small number of demonstrations, the solver needs to figure out the relevant concept and apply it to the test input grid. In particular, the solver must *generate* the answer rather than choose from given candidate answers. Moreover, rather than relying on procedurally generated problems, Chollet hand-designed 1,000 tasks, which were used for a competition on the Kaggle website [21]. Four hundred of the tasks were assigned to a "training set," whose purpose is to give the solver a general idea of what kinds of concepts can be used. Four hundred additional tasks were assigned to an evaluation set for solvers to assess their abilities, and the 200 hundred remaining tasks make up a unreleased (hidden) test set. The tasks were carefully designed to capture "core knowledge" [10] and to assess it in a few-shot, generative framework.

The Kaggle ARC competition allowed each competing program to generate three answers for each task. If one of the answers is correct, the program gets credit for solving that task. Using this metric, the top scorer in the competition was correct on about 21% of the hidden test cases; the second-place scorer was correct on about 19%.

5. Concept-Based Evaluations For ARC

As a second illustration of our concept-based evaluation approach, we created new ARC tasks to evaluate the Kaggle competition's second-place winner [22] (whose code was made publicly available). Here we will call this program *ARC-Kaggle2*. To probe this program's understanding of concepts in the ARC domain, we selected a number of ARC training tasks that it answered correctly, and identified the concepts a human might have used to solve them.

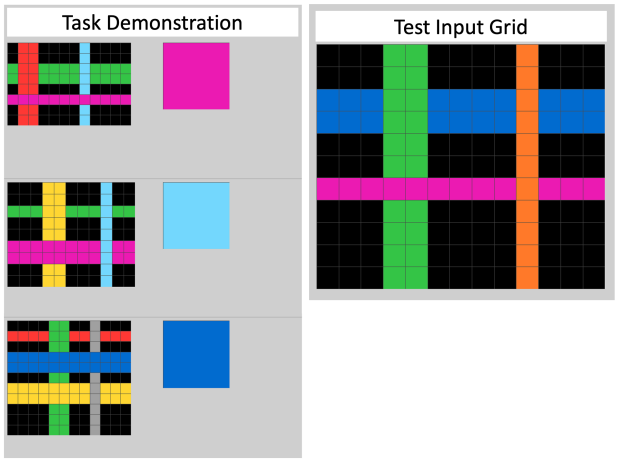
Here we focus on two concepts that appear in the original ARC evaluation set. The first concept involves spatial notions of "top" and "bottom" (or "above" and "below"). The second concept involves the notion of "boundary." Figure 5(a) shows a task from the original ARC evaluation set that focuses on the "top/bottom" concept: The transformation rule is something like "Select the color of the topmost stripe." ARC-Kaggle2 answered this task correctly. Figure 6(a) shows a task from the original ARC evaluation set that focuses on the "boundary" concept: The transformation rule is something like "Move all objects to the red boundary." ARC-Kaggle2 also answered this task correctly.

To probe ARC-Kaggle2's grasp of these two concepts, we created variations on "top/bottom" and 12 variations on "boundary." To give a flavor of these variations, Figures 5(b) and (c) show two of our variants on the "top/bottom" concept, and Figures 6(b) and (c) show two of our variations on the "boundary" concept. [4] Table 2 gives the accuracy (given three guesses per task) of ARC-Kaggle2 on our concept variations. It can be seen that while the program's accuracy on the original ARC test set was 19%, it appears somewhat better on the "top/bottom" concept at 29% correct, and significantly worse on the "boundary" concept at 8% correct. Given the small number of variations we evaluated the system on, we give these results only as an illustration of our concept-evaluation method; a more thorough evaluation would require many more variations.

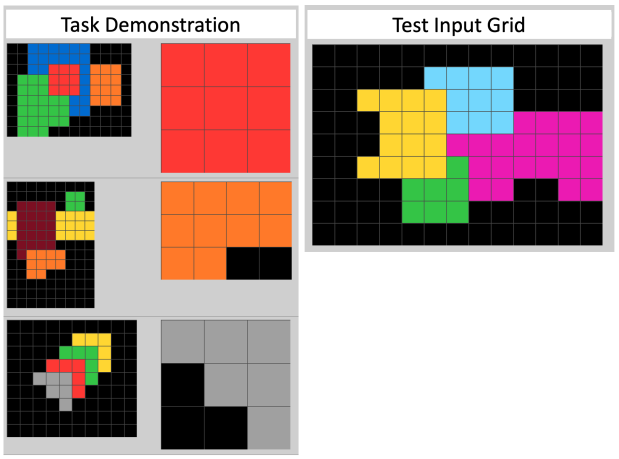
6. Conclusions and Future Work

We have argued for assessing AI abstraction programs using systematic concept-based evaluations rather than random training/test splits or IID test sets. We demonstrated our proposed concept-based evaluation method on existing programs designed to solve problems in the

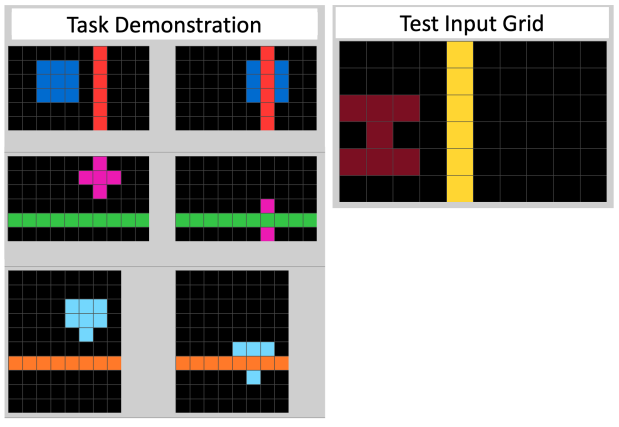
[4] Our ARC task variations can be downloaded from https://melaniemitchell.me/EBeM2022/ARCVariations.zip.



(a)

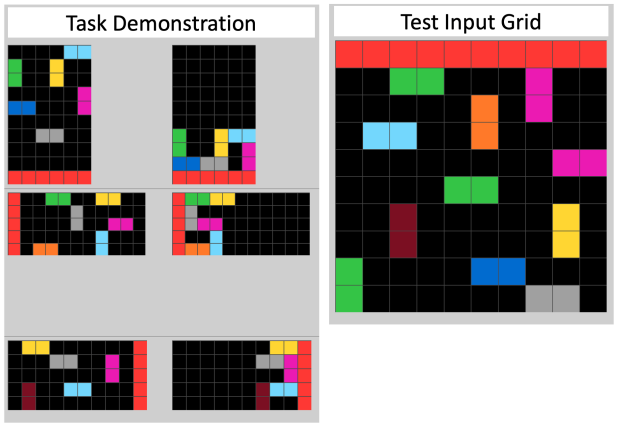


(b)

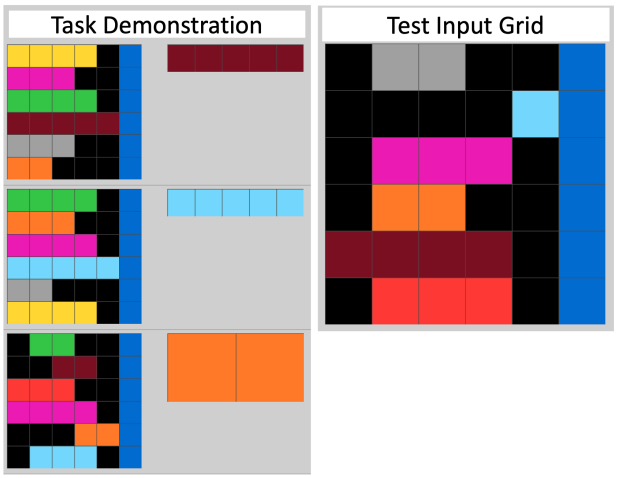


(c)

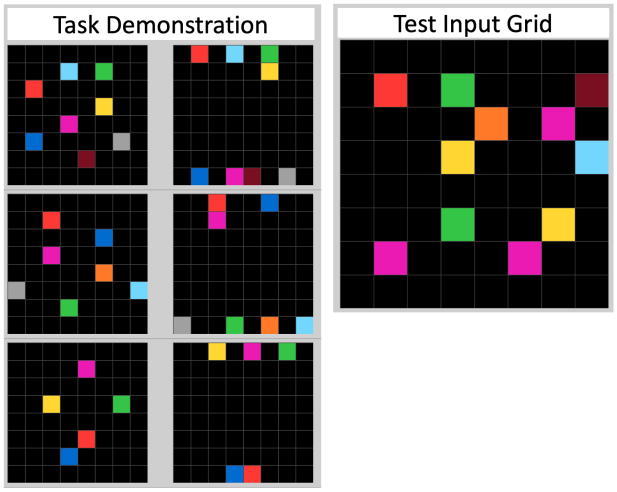
Figure 5: (a) An ARC task (from the original evaluation set) related to the concept of "top" and "bottom"(or "above" and "below"). The transformation rule is something like "extract the color of the topmost stripe." (b) A sample variation on the "top/bottom" concept. The transformation rule is something like "extract the the topmost object." (c) Another sample variation on the "top/bottom" concept. The transformation rule is something like "move the object to below the stripe." Best viewed in color.



(a)



(b)



(c)

Figure 6: (a) An ARC task (from the original evaluation set) related to the concept of "boundary." The transformation rule is something like "Move all objects to the red boundary." (b) A sample variation on the "boundary" concept. The transformation rule is something like "Extract the horizontal stripe that reaches the vertical blue boundary. (c) Best viewed in color. Another sample variation on the "boundary" concept. The transformation rule is something like "Move all objects to their closest outer vertical boundary.

Model	Original ARC Test Set	Top/Bottom Variations (14 tasks)	Boundary Variations (12 tasks)
ARC-Kaggle2	19%	29%	8%

Table 2
ARC-Kaggle2's accuracy on the original ARC test set as well as on our variations on two concepts.

RAVEN and ARC datasets. Our results indicate that evaluation based on accuracy IID tests set can be uninformative in predicting more generalized performance for a given concept. In particular, even for concepts present in problems on which the system did well, its performance on concept variations—meant to probe the system's degree of conceptual *understanding*—can be poor.

The results in this paper are meant as an illustration of the method rather than a thorough evaluation; a more complete evaluation would require assessing the systems on many additional concepts, each explored via numerous problem variations. In the future we plan to develop more thorough concept-based evaluation problem suites in not only the RAVEN and ARC domains but in other idealized abstraction and analogy domains for AI systems (e.g., Bongard problems [23] and letter-string analogies [24]). We also plan to perform human benchmarking studies on these evaluation suites so we can compare human performance with that of machines.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2139983. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. This work was also supported by the Santa Fe Institute.

References

[1] L. W. Barsalou, Challenges and opportunities for grounding cognition, Journal of Cognition 3 (2020).

[2] J. McCarthy, M. L. Minsky, N. Rochester, C. E. Shannon, A proposal for the Dartmouth summer research project on artificial intelligence (First published August 31, 1955), AI Magazine 27 (2006) 12–12.

[3] M. Mitchell, Artificial intelligence hits the barrier of meaning, Information 10 (2019) 51.

[4] W. Landecker, M. D. Thomure, L. M. A. Bettencourt, M. Mitchell, G. T. Kenyon, S. P. Brumby, Interpreting individual classifications of hierarchical networks, in: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2013, pp. 32–38.

[5] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, Nature Machine Intelligence 2 (2020) 665–673.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv (2013) 6199. arXiv:1312.6199.

[7] C. Zhang, F. Gao, B. Jia, Y. Zhu, S.-C. Zhu, RAVEN: A dataset for relational and analogical visual reasoning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, 2019, pp. 5317–5327.

[8] J. C. Raven, J. H. Court, Raven's progressive matrices, Western Psychological Services, 1938.

[9] F. Chollet, On the measure of intelligence, arXiv (2019) 01547. arXiv:1911.01547.

[10] E. S. Spelke, K. D. Kinzler, Core knowledge, Developmental Science 10 (2007) 89–96.

[11] D. G. T. Barrett, F. Hill, A. Santoro, A. S. Morcos, T. Lillicrap, Measuring abstract reasoning in neural networks, in: Proceedings of the International Conference on Machine Learning, ICML, 2018, pp. 4477–4486.

[12] Y. Benny, N. Pekar, L. Wolf, Scale-localized abstract reasoning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12557–12565.

[13] S. Hu, Y. Ma, X. Liu, Y. Wei, S. Bai, Stratified rule-aware network for abstract visual reasoning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 1567–1574.

[14] A. Lovett, K. D. Forbus, Modeling visual problem solving as analogical reasoning, Psychological Review 124 (2017).

[15] S. Spratley, K. Ehinger, T. Miller, A closer look at generalisation in RAVEN, in: European Conference on Computer Vision, Springer, 2020, pp. 601–616.

[16] K. Wang, Z. Su, Automatic generation of Raven's progressive matrices, in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 903–909.

[17] Y. Wu, H. Dong, R. Grosse, J. Ba, The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning, arXiv:2007.04212 (2020).

[18] C. Zhang, B. Jia, F. Gao, Y. Zhu, H. Lu, S.-C. Zhu, Learning perceptual inference by contrasting, Ad-

vances in neural information processing systems 32 (2019).
[19] T. Zhuo, . Kankanhalli, Solving Raven's progressive matrices with neural networks, arXiv:2002.01646 (2020).
[20] M. Małkiński, J. Mańdziuk, Multi-label contrastive learning for abstract visual reasoning, arXiv preprint arXiv:2012.01944 (2020).
[21] F. Chollet, Abstraction and reasoning challenge, 2020. URL: https://www.kaggle.com/c/abstraction-and-reasoning-challenge.
[22] A. de Miquel Bleier, Finishing 2nd in Kaggle's abstraction and reasoning challenge, 2020.
[23] M. M. Bongard, Pattern Recognition, Spartan Books, 1970.
[24] D. R. Hofstadter, M. Mitchell, The Copycat project: A model of mental fluidity and analogy-making, in: K. J. Holyoak, J. A. Barnden (Eds.), Advances in Connectionist and Neural Computation Theory, volume 2, Ablex Publishing Corporation, 1994, pp. 31–112.