MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Are Deep Neural Networks SMARTer than Second Graders?

Cherian, Anoop; Peng, Kuan-Chuan; Lohit, Suhas; Smith, Kevin; Tenenbaum, Joshua B.

TR2023-014 March 28, 2023

Abstract

Recent times have witnessed an increasing number of applications of deep neural networks towards solving tasks that require superior cognitive abilities, e.g., playing Go, generating art, question answering (e.g., ChatGPT), etc. Such a dramatic progress raises the question: how generalizable are neural networks in solving problems that demand broad skills? To answer this question, we propose SMART: a Simple Multimodal Algorithmic Reasoning Task and the associated SMART-101 dataset1, for evaluating the abstraction, deduction, and generalization abilities of neural networks in solving visuo-linguistic puzzles designed specifically for children in the 6–8 age group. Our dataset consists of 101 unique puzzles; each puzzle comprises a picture and a question, and their solution needs a mix of several elementary skills, including arithmetic, algebra, and spatial reasoning, among others. To scale our dataset towards training deep neural networks, we programmatically generate entirely new instances for each puzzle while retaining their solution algorithm. To benchmark the performance on the SMART-101 dataset, we propose a vision-and-language meta-learning model that can incorporate varied state-of- the-art neural backbones. Our experiments reveal that while powerful deep models offer reasonable performances on puzzles in a supervised setting, they are not better than random accuracy when analyzed for generalization – filling this gap may demand new multimodal learning approaches.

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Are Deep Neural Networks SMARTer than Second Graders?

Anoop Cherian[1] Kuan-Chuan Peng[1] Suhas Lohit[1] Kevin A. Smith[2] Joshua B. Tenenbaum[2]
[1]Mitsubishi Electric Research Labs (MERL) [2]Massachusetts Institute of Technology (MIT)
{cherian, kpeng, slohit}@merl.com {k2smith, jbt}@mit.edu

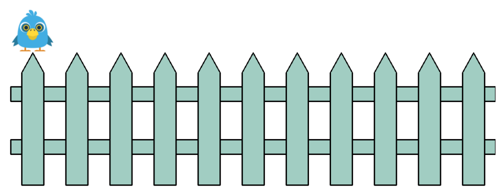
Abstract

Recent times have witnessed an increasing number of applications of deep neural networks towards solving tasks that require superior cognitive abilities, e.g., playing Go, generating art, question answering (e.g., ChatGPT), etc. Such a dramatic progress raises the question: how generalizable are neural networks in solving problems that demand broad skills? *To answer this question, we propose SMART: a* Simple Multimodal Algorithmic Reasoning Task *and the associated SMART-101 dataset*[1]*, for evaluating the abstraction, deduction, and generalization abilities of neural networks in solving visuo-linguistic puzzles designed specifically for children in the 6–8 age group. Our dataset consists of 101 unique puzzles; each puzzle comprises a picture and a question, and their solution needs a mix of several elementary skills, including arithmetic, algebra, and spatial reasoning, among others. To scale our dataset towards training deep neural networks, we programmatically generate entirely new instances for each puzzle while retaining their solution algorithm. To benchmark the performance on the SMART-101 dataset, we propose a vision-and-language meta-learning model that can incorporate varied state-of-the-art neural backbones. Our experiments reveal that while powerful deep models offer reasonable performances on puzzles in a supervised setting, they are not better than random accuracy when analyzed for generalization – filling this gap may demand new multimodal learning approaches.*

1. Introduction

> "*An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.*"
>
> *The Dartmouth Summer Project on AI, 1956*

Deep learning powered AI systems have been increasing in their data modeling abilities at an ever more vigor in the recent times, with compelling applications emerging frequently, many of which may even seem to challenge human abilities. A few notable such feats include but are not limited to game playing (*e.g.*, AlphaGo [60]), language-guided image generation (*e.g.*, the recent DALLE-2 [54] and ImageGen [56]), creative story writing (*e.g.*, using GPT-3 [10]), solving university level math problems [17], algorithmic inference [20], and general question-answering/dialog (*e.g.*, ChatGPT [48] and variants). Such impressive performances have prompted an introspection into the foundation of what constitutes artificial intelligence and deriving novel tasks that could challenge deep models further [13, 37, 45, 55].



Question: *Bird Bobbie jumps on a fence from the post on the left end to the other end. Each jump takes him 4 seconds. He makes 4 jumps ahead and then 1 jump back. Then he again makes 4 jumps ahead and 1 jump back, and so on. In how many seconds can Bobbie get from one end to the other end?*
Answer Options: A: 64 B: 48 C: 56 D: 68 E: 72

Figure 1. An example puzzle instance from our SMART-101 dataset generated using our programmatic augmentation method. Solving this puzzle needs various skills such as *counting* the number of posts, *spatially* locating *Bobbie*, and using the *details in the question* to derive an algorithm for the solution. At a foundational level, a reasoning agent needs to recognize abstracted objects such as posts and identify the *bird*. The answer is shown below[2].

While deep neural networks offer compelling performances on specialized tasks on which they are trained on, (i) how well do they model abstract data, attend on key entities, and transfer knowledge to solve new problems? (ii) how fluid are they in acquiring new skills? and (iii) how effective are they in the use of language for visual reasoning? We task ourselves to understand and seek a way to answer these

[1]The SMART-101 dataset is publicly available at: https://doi.org/10.5281/zenodo.7761800

[2]The answer to the puzzle in Figure 1 is: C.

questions for state-of-the-art (SOTA) vision and language deep learning models. An approach that has been taken several times in the past is to design specialized datasets that can measure the cognitive abilities of well-trained neural networks. For example, in CLEVR [34], a diagnostic dataset is proposed that comprises visuo-linguistic spatial reasoning problems. The abstraction abilities of neural networks have been explored towards solving types of Bongard problems [33, 47] and human IQ puzzles (e.g., Ravens progressive matrices) have been extended to evaluate neural reasoning abilities [7, 8, 31, 49, 64, 66, 72, 75]. However, while the puzzles in these prior works are often seemingly diverse, they are often confined to a common setting and may need only specialized skill sets, bringing in inductive biases that could be exploited by well-crafted deep learning models, thereby solving such puzzles with near perfect accuracy [59, 64].

In this paper, we take a look back at the foundations of intelligence, by asking the question: *Are state-of-the-art deep neural networks capable of emulating the thinking process of even young children?* To gain insights into answering this question, we introduce the Simple Multimodal Algorithmic Reasoning Task (SMART) – a visuo-linguistic task and the associated SMART-101 dataset built from 101 distinct children's puzzles. As this is the first step in this direction, we keep the puzzles simple – to ensure this, we took inspiration from the puzzles in the Math Kangaroo USA Olympiad [3] which has puzzle sets professionally designed for children in the age group of 6–8. Each puzzle in our dataset has a picture describing the problem setup and an associated natural language question. To solve the puzzle, one needs to use the question to gather details from the picture and infer a simple mathematical algorithm that leads to a solution to be matched against multiple answer options. In Figure 1, we illustrate our task with an example puzzle from our dataset. Unlike prior datasets with similar goals, each of the 101 puzzles in our dataset is distinct and needs a broad range of elementary mathematical skills for their solutions, including skills in algebra, basic arithmetic, geometry, ordering, as well as foundational skills to interpret abstract images, and execute counting, spatial reasoning, pattern matching, and occlusion reasoning. To the best of our knowledge, this is the first dataset that offers such a richly diverse set of visuo-linguistic puzzles in an open-world setting, with a psychometric control on their difficulty levels against human performance.

To benchmark performances on the SMART-101 dataset, we propose an end-to-end meta-learning based neural network [21], where we use a SOTA pre-trained image encoder backbone (*e.g.*, Transformers/ResNets) to embed the picture part of the puzzles, and a strong large language model (*e.g.*, GPT/BERT) to model the questions. As each puzzle may have a different range for their answers (*e.g.*, selection from a few choices, sequential answers, *etc.*), we propose to treat each puzzle as a separate task, with task-specific neural heads and training objectives, while a common vision-language backbone is used on all the puzzles.

We provide experiments using our learning framework under various evaluation settings, analyzing the ability of SOTA vision and language backbones for: (i) in-distribution generalization, when training and test data are from the same distributions of puzzle instances, and out-of-distribution generalization, when training and test data are from: (ii) distinct answer distributions, or (iii) different puzzles. We find the backbones performing poorly in our model on (i) and (ii), while failing entirely on (iii), suggesting that solving our dataset would demand novel research directions into algorithmic reasoning.

We experiment on various settings, evaluating the ability of our model to (i) solve puzzles when trained and tested on the same distribution of instances, (ii) out of distribution generalization when training and testing data are disjoint at the answer level, and (iii) out of distribution generalization when the training and testing sets are disjoint at the puzzle levels. We find that our model performs poorly on the tasks (i) and (ii), while failing entirely on (iii), suggesting that solving our dataset would demand novel research directions into neural abstractions, and algorithmic reasoning abilities.

We summarize below the key contributions of this paper.

1. With the goal of making progress towards improving the visuo-linguistic algorithmic reasoning abilities of neural networks, we introduce a novel task: SMART, and the associated large-scale SMART-101 dataset.
2. We propose a programmatic augmentation strategy for replicating abstract puzzles.
3. We design a baseline meta-solver neural architecture for solving the puzzles in our task.
4. We present experiments using our approach in various algorithmic generalization settings, bringing out key insights on the performance of SOTA neural networks on this task. We also compare performances against humans and using large language models.

2. Related works

To set the stage, we briefly review below a few prior methods and datasets proposed towards understanding the reasoning abilities of deep neural networks.

Solving IQ puzzles: via creating computer programs has been a dream since the early days of exploration into AI [28, 43, 44]; Evan's ANALOGY [19] and Hofstader's CopyCat, among others [30] are famous tasks in this direction. With the resurgence of deep learning, there have been several attempts at re-considering such puzzles, with varied success. In Table 1, we briefly review such tasks and datasets (see Małkiński and Mańdziuk [42] for an in-depth survey). While, the goal of these works have been

Dataset	Involve language	Dataset size	Task nature
Bongard-LOGO [47]	✗	12K	few-shot concepts, abstract shape reasoning
Bongard-HOI [33]	✗	53K	few-shot concepts, human-object interaction
ARC [13]	✗	800	generate image based on abstract rules
Machine Number Sense [74]	✗	280K	solving arithmetic problems
RAVEN [72]	✗	70K	finding next image in sequence
Image riddles [4]	✓(fixed question)	3333	finding common linguistic descriptions
VLQA [57]	✓(variable questions)	9267	spatio-temporal reasoning, info lookup, mathematical, logical, causality, analogy, *etc*.
PororoQA [36]	✓(variable questions)	8913	reason from cartoon videos about action, person, abstract, detail, location, *etc*.
CLEVR [34]	✓(variable questions)	100K	exist, count, query attributes, compare integers/attribute
SMART-101 (ours)	✓(variable questions)	200K	8 predominant algorithmic skills and their compositions (see Figure 2)

Table 1. Comparison between our SMART-101 dataset with existing datasets related to visual reasoning.

towards capturing human cognition through machine learning models, their tasks are often specialized and when provided enough data, the neural networks apparently leverage shortcomings in the dataset towards achieving very high accuracy [28, 64, 73], defaulting the original goals.

Neuro-symbolic learning and program synthesis: approaches consider solving complex tasks via decomposing a scene into entities and synthesizing computer programs that operate on these entities; thereby plausibly emulating human reasoning. The DreamCoder approach [18] for program synthesis to draw curves, solving Bongard problems using program induction [63], solving Raven's matrices using neuro-symbolic methods [29], and Bongard LOGO [47] are a few recent and successful approaches towards neuro-algorithmic reasoning, however, their generalization to tasks beyond their domains is often unexplored.

Visual and language: tasks for understanding and reasoning on natural images [5, 6, 32, 34, 51] have been very successful using deep neural networks, lately [9, 35, 39, 41, 51, 58, 61, 62, 65, 67, 70, 71]. Similar to such tasks, our goal in SMART-101 is to jointly interpret vision and language modalities for solving various reasoning tasks. However, different from such approaches, our images are not necessarily natural images, instead are mostly sketches without textures; thereby avoiding the unexpected and implicit inductive biases.

Understanding children's cognition: for solving a variety of age-appropriate problems has been intensively studied over the years [14, 23, 37] via studying their ability to form abstract, hierarchical representations of the world, acquire language and develop a theory of mind [22]. A particularly useful and common approach to understanding children's cognitive abilities, albeit imperfectly, is to present them with puzzles such as those in IQ tests [38, 46, 68]. To the best of our knowledge, it is the first time that a dataset has been built in this direction, that can allow exploration of generalized reasoning abilities at a level of children's cognition, and that can be potentially useful not only in computer vision, but also for studying a breadth of abilities spanning psychology, neuroscience, and cognitive science.

3. Proposed approach

3.1. Task and the SMART-101 dataset

As alluded to above, our goal is to understand the abilities and shortcomings of SOTA deep models for visuo-linguistic reasoning. With this goal in mind, we propose the Simple Multimodal Algorithmic Reasoning Task and the SMART-101 dataset, consisting of visuo-linguistic puzzles in a multiple-choice answer selection setting.

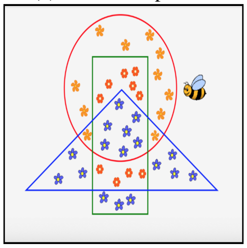
Each puzzle in SMART-101 consists of an image I, a natural language question Q, and a set of five multiple choice answers $\mathcal{A}$, and the task is to have an AI model f_θ, parameterized by θ, that can provide the correct answer a to a given problem tuple $(I, Q, \mathcal{A})$, *i.e.*,

$$f_\theta(I, Q) \rightarrow a \in \mathcal{A}. \quad (1)$$

To learn the parameters θ of the model f_θ, we use a dataset $\mathcal{R} = \{\pi_1, \pi_2, \cdots, \pi_K\}$ consisting of a set of $K = 101$ distinct puzzles. We call each π a *root puzzle*. To train deep learning models, we need large datasets, and to this end, we create new non-identical puzzle instances for each root puzzle. That is, for each $\pi \in \mathcal{R}$, we programmatically produce $\mathcal{P}_\pi = \left\{p_1^\pi, p_2^\pi, \cdots, p_{n_\pi}^\pi\right\}$, where p^π denotes a new instance of root puzzle π. Thus, our full dataset $\mathcal{D} = \cup_{\pi\in\mathcal{R}}\mathcal{P}_\pi$.

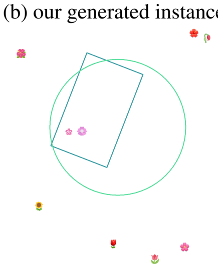
To choose the root puzzles, one may consider a variety of sources, *e.g.*, puzzle books, IQ tests, online resources, *etc*. In this work, we derived them from the Math Kangaroo (MK) USA Olympiad [3], which is an annually held mathematical competition meant for kids from first to tenth grade. For this paper, we selected problems designed for children of ages 6–8 (typically first and second graders). Given that MK is a professionally-held competition, it contains high quality content with significant diversity in children's skills needed for solving the puzzles and offer careful categorization of the algorithmic complexity/difficulty needed for solving them. Table 2 shows some example root puzzles from our SMART-101. Further, and most importantly, the puzzles being part of a competition, helps gather statistically significant scores on children's performances, which is perhaps difficult to obtain otherwise.

(a) MK's root puzzle



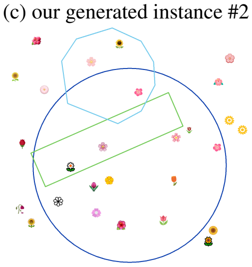
Question: A bee collected pollen from all the flowers inside the rectangle but outside the triangle. From how many flowers did the bee collect pollen? **Options**: A: 9, B: 10, C: 13, D: 17, E: 20

(b) our generated instance #1



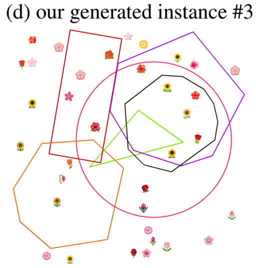
Question: We want to pick up all the flowers that are inside the rectangle and inside the circle simultaneously. How many flowers should we pick up? **Options**: A: 5, B: 6, C: 2, D: 1, E: 3

(c) our generated instance #2

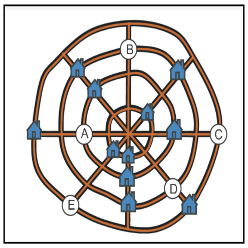


Question: We want to pick up all the flowers that are inside the circle but outside the rectangle simultaneously. How many flowers should we pick up? **Options**: A: 7, B: 14, C: 15, D: 9, E: 11

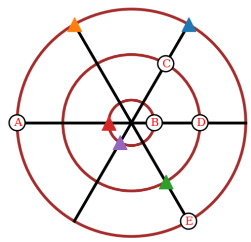
(d) our generated instance #3



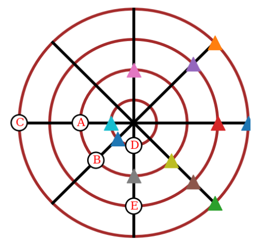
Question: All the flowers that are outside both the circle and triangle simultaneously are picked up. The number of flowers which are picked up is: **Options**: A: 27, B: 24, C: 26, D: 29, E: 23



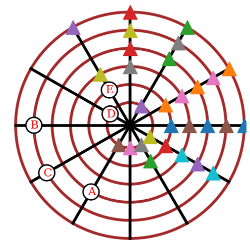
Question: A village with 12 houses has four straight roads and four circular roads. The map shows 11 of the houses. On each straight road there are 3 houses. On each circular road, there are also 3 houses. Where on the map should the 12th house be put? **Options**: A, B, C, D, E



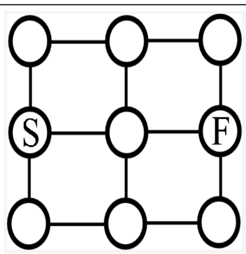
Question: A town with 6 houses has 3 straight pathways and 3 circular pathways. The image shows 5 of the houses. On each straight pathway there are 2 houses. On each circular pathway, there are also 2 houses. Which location on the image should the 6th house be built? **Options**: A, B, C, D, E



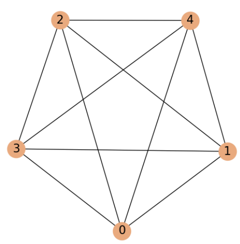
Question: A small town with 12 huts has 4 straight lanes and 4 circular lanes. The map depicts 11 of the huts. On each straight lane there are 3 huts. On each circular lane, there are also 3 huts. Which location on the map should the 12th hut be put? **Options**: A, B, C, D, E



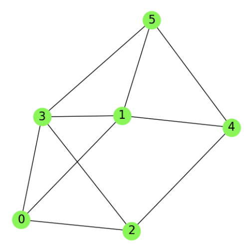
Question: A community with 30 condos has 6 straight paths and 6 circular paths. The picture illustrates 29 of the condos. On each straight path there are 5 condos. On each circular path, there are also 5 condos. Which place on the picture should the 30th condo be added? **Options**: A, B, C, D, E



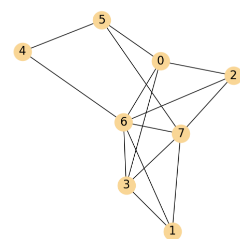
Question: In one jump, Jake jumps from one circle to the neighboring circle along a line, as shown in the picture. He cannot jump into any circle more than once. He starts at circle S and needs to make exactly 4 jumps to get to circle F. In how many different ways can Jake do this? **Options**: A: 3, B: 4, C: 5, D: 6, E: 7



Question: In one jump, Pamela jumps from one circle to the neighboring circle along a line, as shown in the picture. She cannot jump into any circle more than once. She starts at circle 2 and needs to make exactly 4 jumps to get to circle 0. In how many different ways can she do this? **Options**: A: 6, B: 11, C: 2, D: 10, E: 0



Question: In one jump, Louis jumps from one circle to the neighboring circle along a line, as shown in the picture. He cannot jump into any circle more than once. He starts at circle 4 and needs to make exactly 2 jumps to get to circle 1. In how many different ways can Louis do this? **Options**: A: 3, B: 2, C: 0, D: 1, E: 6



Question: In one jump, Chris jumps from one circle to the neighboring circle along a line, as shown in the picture. He cannot jump into any circle more than once. He starts at circle 1 and needs to make exactly 7 jumps to get to circle 6. In how many different ways can Chris do this? **Options**: A: 10, B: 8, C: 7, D: 2, E: 1

Table 2. Examples of the root puzzles (left) from the Math Kangaroo Olympiad [3] and our generated puzzle instances, belonging to categories: counting (top), logic (middle), and path tracing (bottom). The answer is marked in red.

3.2. Programmatic puzzle augmentation

In this subsection, we detail our approach to replicate a root puzzle into its diverse instances; potentially expanding the dataset to a size that is large enough for adequately training deep neural networks. While, one may resort to standard data augmentation methods (such as cropping, rotations, *etc.*) to produce data from the root puzzles, such an approach may be unsuitable, because: (i) such operations may make the problem invalid, *e.g.*, flipping an image to augment it might make a question on the orientation of an object incorrect, and (ii) such augmentations might not change the puzzle content much, *e.g.*, rotating an image of a circle. A different direction is perhaps to create more puzzles via human help, *e.g.*, Amazon Turkers. However, this will need specialized creative skills that could be difficult to obtain and can be expensive.

Intuitively, as we are seeking a model to learn an underlying *algorithm* for solving the puzzles, we should consider puzzle augmentations that make a model algorithmically-equivariant to their solutions. Inspired by this insight, we propose to programmatically augment the puzzles via remaking a root puzzle using a computer program and randomly changing the program settings to diversify the puzzles. Specifically, as our goal is for a reasoning method to learn an "*algorithm*" to solve a puzzle (rather than using only the perception modules), we randomly change the visual, lingual, and contextual puzzle attributes using content from a variety of domains, thereby bringing in significant diversity in each recreated puzzle instance. To accomplish this, the new puzzle images are sampled from varied sources, *e.g.*, the Icons-50 dataset [27], random internet cliparts, *etc*., and their spatial organizations, colors, textures, shapes, etc. are all randomly-sampled.

While the above approach for puzzle augmentation seems straightforward, it needs to be noted that to replicate each root puzzle, sometimes special expertise is needed to produce suitable images, the associated questions, and produce answers that are correct. To illustrate this intricacy, in Table 2, we illustrate three puzzles and their augmentations using our approach. Below, we provide details of their augmentation programs.

Table 2 Row 1. We first randomly sample two different types of shapes s_1 and s_2 from a shape set, with random spatial locations and sizes. Optionally, we also including distractors. Second, we randomly sample the flower instances from the Icons-50 dataset [27] and paste them to the images such that the boundaries of s_1 and s_2 do not intersect with those of the icons. Third, we randomly sample the relationship associated with s_1 and s_2 from $\{$inside, outside$\}$ to create the question and compute the answer.

Table 2 Row 2. For a problem setting with n circles (and roads), the replication of this puzzle amounts to finding an $X = \begin{bmatrix} X_{11}, X_{12} \\ X_{21}, X_{22} \end{bmatrix}$, where $X_{11} = X_{22}$ and $X_{12} = X_{21}$ with X_{ij}'s being $n \times n$ integer matrices under the constraint that their rows and columns sum to k (the number of houses in the puzzle). This problem is cast as an integer programming problem and solved using the GLPK toolkit [1] for random puzzle attributes.

Table 2 Row 3. We sample the number of nodes N from $[4, N_{max}]$, and sample random graphs with number of edges in $\left[N, \frac{N(N-1)}{2}\right]$. We use the NetworkX Python package [2, 24] for rendering random graphs, post which we randomly sample source and target nodes to generate a question. Next, we find all simple paths between the vertices, compute their lengths, and choose one target path in the generated question to form the correct answer.

3.3. Details of the dataset

We categorize the 101 root puzzles in the SMART-101 dataset into eight different classes based on the type of basic skill needed to solve them, namely: (i) variants of counting (*e.g.*, counting lines, basic shapes, or object instances), (ii) basic arithmetic (*e.g.*, simple multiplication), (iii) logical reasoning (*e.g.*, *Is X taller than Y but shorter than Z?*), (iv) algebra (*e.g.*, *Is the sum of the sides of a cube X?*), (v) spatial reasoning (*e.g.*, *Is X behind Y?*), (vi) pattern finding (*e.g.*, *If the pattern in X is repeated, which point will it pass through?*), (vii) path finding (*e.g.*, *which option needs to be blocked so that X will not reach Y in a maze?*), and (viii) measurement (*e.g.*, *for a grid X if each cell is 1 cm, how long is X?*). In Figure 2, we show the distribution of puzzles in SMART-101 across these classes.[3]

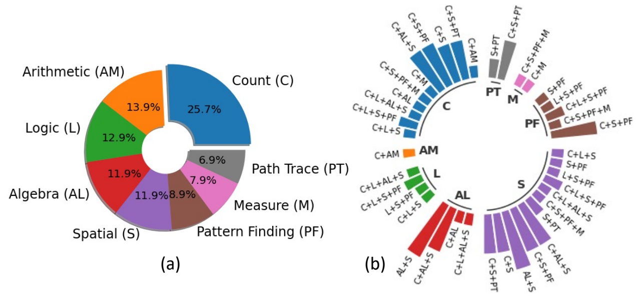


Figure 2. Analysis of the various statistics of problems in the SMART-101 dataset. (a) shows the distribution of problems among the eight classes of predominant math skills needed to solve them. In (b), we plot the composition of various skills that are potentially needed to solve a problem.

As one can see from the sample puzzles provided in Table 2, it is not just the above skills that one needs to solve them, instead their solution demands a composition of the above skills. For example, to solve the puzzle in the first row of Table 2, one needs to recognize the *pattern* for similar flowers, *spatially reason* whether each flower is within or outside a given shape, and *count* the flowers. The class distribution in Figure 2(a) characterizes the basic skill needed (*e.g.*, counting) to solve this problem, and might not provide the full skill diversity. Thus, in Figure 2(b), we provide a more comprehensive analysis of the various compositions of skills needed to solve the of problems in SMART-101. As is clear from this pie chart, each puzzle in our dataset demands a multitude of skills – attesting to the complexity of the task and the challenge it offers.

Question Augmentation. To create new questions for puzzle instances, we follow a combination of three different strategies: (i) for puzzle questions with numbers, we replace them with new numbers sampled from a range, (ii) replace the sentence structure with manually-generated templates, and (iii) use slotted words in the template, where the words in the slots are sampled from potential synonyms, while ensuring the question is grammatically correct, sensible, and captures the original goal and difficulty of the puzzle.

4. SMART-101 reasoning model

Each puzzle in SMART-101 has distinct problem characteristics and diverse ranges for their answers (*e.g.*, numeric, alphabets, sentences, and words); thus, using a single loss

[3]Note that this categorization was done among the authors via a manual categorization and voting on the root puzzles.

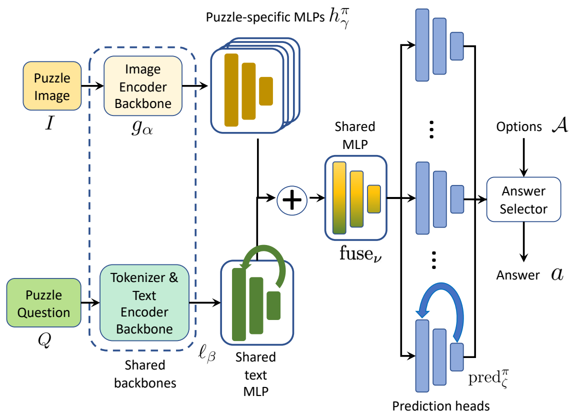


Figure 3. An illustration of our learning and reasoning model.

for all puzzles may be sub-optimal. While, one may resort to multi-task learning (MTL), however having samples from all puzzles to train in MTL may need large batches that can be difficult to scale. Further, we desire our model to be trained and evaluated in few-shot settings. A natural way to resolve all the above challenges is to consider a meta-learning architecture [21], pictorially described in Figure 3.

Mathematically, let g_α and ℓ_β be the *image backbone* and the *language backbone* (combined with an RNN to aggregate the word embeddings) shared across all the puzzles in $\mathcal{D}$ respectively, where α and β capture their parameters. As distinct root puzzle images have specific characteristics for the solution (e.g., some of the images have their answer options embedded within the image), we found it useful to have a puzzle-specific image head. To this end, we attach a small (2-layered) multi-layer perceptron (MLP), denoted h_γ^π, to the output of the image backbone, where h_γ^π is specific to each root puzzle π and has its own parameters γ. Using these modules, our prediction model for puzzle π is:

$$f_\theta^\pi(I, Q) := \operatorname{pred}_\zeta^\pi(\operatorname{fuse}_\nu((h_\gamma^\pi(g_\alpha(I)) + \ell_\beta(Q))), \quad (2)$$

where fuse_ν denotes a shared MLP to fuse the image and language embeddings and $\operatorname{pred}_\zeta^\pi$ is a puzzle-specific prediction head that maps a given puzzle tuple to the domain of the puzzle answers (with its own parameters). For example, a puzzle answer may be a sequence, for which $\operatorname{pred}_\zeta^\pi$ would be an RNN, while for another puzzle, the response could be an integer in 1–100, for which $\operatorname{pred}_\zeta^\pi$ could be an MLP classifier with 100 softmax outputs. We abstractly represent trainable parameters in various modules by θ.

To train the model in Eq. (2), we optimize:

$$\min_\Theta \mathbb{E}_{\pi\sim\mathcal{R}} \mathbb{E}_{(I,Q,a)\sim\mathcal{P}_\pi} \operatorname{loss}_\pi\big(f_\theta^\pi(I, Q) - a\big), \quad (3)$$

where $\Theta = \cup_{\pi\in\mathcal{R}} \{\theta\}_\pi$ and loss_π is a puzzle-specific loss that is activated based on the root puzzle π for an instance (I, Q, a) in a given batch. Specifically, we sample the tasks (puzzles) and instances from those tasks to form mini-batches to train the puzzle-specific heads for several iterations, followed by combining the gradients from the tasks to update the backbones through the puzzle heads, as in [21]. Note that a is the correct answer and loss_π could be: (i) a softmax cross-entropy loss (selecting in a discrete answer range) or (ii) an ℓ_1 regression loss predicting a scalar value.

At inference, we select the answer from the options as:

$$\hat{a} = \arg\max_{\alpha\in\mathcal{A}} \operatorname{sim}_\pi\big(f_\theta^\pi(I, Q), \alpha\big), \quad (4)$$

where sim_π captures the similarity of a predicted answer value against the choices in $\mathcal{A}$, and sim_π is specific to the problem π (*e.g.*, euclidean distance for numerals).

5. Experiments

In this section, we detail the experimental protocol to evaluate the models for solving SMART-101.

5.1. Data splits

We propose four different data splits that evaluate varied generalization properties of an method/model to solve SMART-101. The splits are: (i) **Puzzle Split** (PS) with the goal to evaluate extreme generalization. In this setting, we split the root puzzles into 77-3-21 (train-val-test).[4] The performance is evaluated on the test set consisting of puzzles that the model has never seen during training (as a zero-shot solver). (ii) As PS is perhaps extremely challenging for today's machine learning approaches, we include a **Few-shot Split** (FS), where the model sees m $(= 100)$ instances from all the 21 puzzles used as the test set in PS. (iii) **Instance Split** (IS) evaluates the in-distribution performance of a model (supervised learning). For IS, we split all the instances of all root puzzles into 80-5-15 (%). IS receives puzzle-specific information on all puzzles and is the easiest setting for a model to perform. (iv) **Answer Split** (AS) that evaluates the generalization to answers that a model has not seen during training. In this split, we compute the distribution of all answers (a in Eq. 3) across instances for a root puzzle, find the median answer, and remove all instances that have this median answer from the training set; these instances are used only during inference.

5.2. Evaluation

We use two metrics to evaluate performance: (i) the solution accuracy S_{acc} that computes the frequency with which the correct *solution* was produced by a model and (ii) the option selection accuracy O_{acc} that measures the frequency with which the correct *option* was selected by a model. To clarify, for the root puzzle in Table 2 Row 1, let us say a

[4]In PS test, we use 2 counting, 5 logic, 4 algebra, 1 path, 1 measurement, 4 spatial, 3 arithmetic, and 1 pattern puzzles.

model produced an answer 8. Since 8 is not in the option set, the closest option 9 will be selected, *i.e.*, the correct option will be selected even if the wrong answer is produced. In this case, its O_{acc}=100%. while its S_{acc}=0%.

5.3. Backbone models

We evaluate popular pretrained image, language, and vision-and-language backbones[5] using the reasoning architecture in Figure 3; see the extended paper [12] for details.

Image Backbones. We consider three groups of models: (i) ResNets, (ii) Transformers, and (iii) contrastively pretrained models. For (i), we use ResNet-50 and ResNet-18 [26]. For (ii) we use several variants, including Vision-Transformers (ViT) [16], Swin-Transformers [40] (Swin-T and Swin-B) and Cross-Transformers [69]. While we finetune ViT and Swin-Ts from pre-trained models, we train Cross-Transformers from scratch on our dataset. For self-supervised pre-trained models, we use SimSiam [11] based on ResNet-50 and Masked Autoencoders [25](MAE).

Language Backbones. As alluded to above, we use either a learned feature embedding (Emb.) for encoding the questions (using a vocabulary of ~7K words created on SMART-101) or a SOTA embedding model and its associated tokenizer. We consider 3 text embedding models: (i) GPT-2 [53], (ii) BERT [15], and (iii) GloVe [50].

Vision-and-Language Models. We also consider multimodal pre-trained models that are specifically trained for aligning vision with language. In this setting, we consider the recent CLIP [52] and FLAVA [61].

5.4. Experimental Results

In Table 3, we present our results using our reasoning framework and varied backbones on both S_{acc} and O_{acc} metrics, and against human performance.

Second Grader Performance: The main goal of this paper is to gauge the performance of SOTA deep neural networks against those of second-graders. In Table 3, we report averaged category-wise performances of children (in grades 1 and 2) who participated in the Math Kangaroo competition (see [12] for details). Overall, children perform at nearly 77% average accuracy on all the 21 PS puzzles.

Baselines: To ensure that SMART-101 answer options are devoid of any biases, we report two baseline performances that do not involve any learning, namely: (i) *greedy*, that selects the most frequent answer from the training set instances for each root puzzle, and (ii) *uniform*, that randomly samples an answer. Table 3 shows that O_{acc} for all the baseline methods is nearly 20%, suggesting that our answer options are uniformly distributed among the five choices.

Supervised Learning (IS) Performances: For these experiments, we use the learned word embeddings (Emb.). Surprisingly, we find that in IS, ResNet models (R18/R50/SimSiam) perform significantly better than most Transformer models on average (Table 3-IS). To ensure this is not an implementation artifact, we repeat our experiments either via training the models from scratch (Cross-Transformers) or fine-tuning pretrained models (Swin-B, Swin-T, ViT-16, and MAE). These models offer varied amounts of global and local self-attention for reasoning. Table 3 shows that most Transformer variants we compare to do relatively well in *Arithmetic* (~40% on S_{acc} for ViT-16, ~34% for Swin-T and MAE, *etc.*), while they perform the least on tasks that need path tracing. We find that prtrained vision-and-language models (FLAVA and CLIP) perform slightly better than Transformers and show improved performances on counting, logic, and pattern finding. Using R50 image backbone, we further evaluate the performances against various language model choices. We find in Table 3-IS that richer (pretrained) language models such as GloVe, GPT2 or BERT improve the performance over Emb., with benefits in almost all puzzle categories.

Analysis of Generalization: The fundamental goal of this paper is to understand the generalization abilities of SOTA deep models. In Table 3 (under Puzzle Split), we report results analyzing extreme generalization using Transformers, CLIP, and FLAVA. In these experiments, we used the publicly available pretrained backbones and trained only puzzle heads. From the table, we find that SOTA models fail entirely, often selecting a random answer (O_{acc}~20%). We also evaluate our best setting (R50 + BERT) via fine-tuning (FT) R50 with classification (Cls.) and regression (Reg.) losses; however, without any improvement.

To ameliorate extreme generalization, we explore the few-shot (FS) setting where the model is shown m instances of a puzzle during training that is otherwise hidden in the PS split. Even for an $m = 100$, Table 3 (FS) shows that the S_{acc} improves by nearly 6% (from $\sim$ 10% in PS to $\sim$ 16% in FS), suggesting that the model has perhaps learned several useful embeddings and may learn new skills quickly. Next, using R50 + BERT, in classification and regression settings, we evaluate answer generalization (on AS split). Table 3 (AS) shows our classification model fails entirely on AS (0% on S_{acc}). This is unsurprising as on the AS split, the deep model is masked from seeing a particular answer, which is used only during testing. However, Table 3 (FS) also shows that using regression allows the model to interpolate the seen answers, leading to an S_{acc} of 16.3%.

Ablation studies: Table 4 reports the ablations on puzzle-specific image heads and meta-learning as against multi-task learning (MTL). As is expected, when adding the puzzle heads, the performance improves. We find that using meta-learning is important and leads to a dramatic (~12%) improvement in performance. Our results also confirm that both vision and language are essential to solve SMART-101.

[5]All the pretrained backbone models are downloaded from public repositories, specifically https://huggingface.co/models.

Puzzle Category →	Count	Arithmetic	Logic	Path Trace	Algebra	Measure	Spatial	Pattern Finding	Average
Puzzle Split (PS) – Extreme Generalization Experiments									
Avg. 2^{nd} Grader Performance	**72.8**	**81.3**	**82.2**	**81.1**	**64.5**	**90.4**	**74.8**	**88.6**	**77.1**
Greedy (baseline)	19.1/21.4	14.0/21.4	18.5/21.1	21.8/21.1	13.5/21.5	23.1/20.9	18.2/21.2	21.4/21.4	17.7/21.3
Uniform (baseline)	7.74/20.0	8.00/20.0	7.61/20.0	18.9/20.0	6.94/20.0	5.62/20.0	14.2/20.0	20.0/20.0	11.20/20.0
MAE + BERT	5.89/19.1	5.24/26.7	5.23/25.9	0.0/0.0	8.34/17.9	0.0/0.0	2.85/10.6	0.0/0.0	4.74/17.4
SimSiam + BERT	6.44/18.3	7.14/22.4	6.56/27.0	6.54/18.5	3.62/24.5	12.1/26.7	14.8/23.4	0.0/21.2	8.09/23.8
Swin_T + BERT	8.02/12.5	3.14/20.1	10.1/23.9	17.1/20.4	6.77/21.3	11.1/21.9	12.4/17.6	21.3/21.3	9.49/20.2
ViT-16 + BERT	9.41/22.7	5.77/26.8	6.95/25.1	4.72/18.7	5.57/15.1	8.68/21.3	11.6/21.5	18.9/19.7	8.51/21.6
CLIP	9.04/17.4	3.70/19.9	8.50/25.9	25.9/25.9	8.36/30.0	9.42/22.9	15.9/21.7	22.9/22.9	11.9/24.1
FLAVA	12.0/29.6	2.90/18.4	7.27/28.7	0.0/0.0	1.99/28.2	0.0/0.0	6.70/11.6	0.0/0.0	7.35/25.2
R50 + BERT (FT + Cls.)	10.9/18.3	6.96/15.8	12.8/20.8	19.6/19.7	7.95/15.1	16.9/26.7	13.4/17.7	0.0/21.2	11.7/18.9
R50 + BERT (FT + Reg.)	12.0/22.8	5.08/21.3	4.24/16.2	18.4/18.4	4.89/22.2	15.1/25.9	11.9/17.9	19.0/19.0	8.21/19.7
Few-Shot Split (FS) Experiments									
R50 + BERT (Cls.)	24.5/33.2	15.6/23.3	23.8/28.8	0.0/0.0	13.2/25.1	0.0/0.0	10.2/15.2	0.0/0.0	15.1/21.8
R50 + BERT (Reg.)	19.8/33.6	13.9/26.3	18.2/26.9	18.7/18.7	10.3/24.4	11.6/25.8	20.8/29.8	21.9/22.3	16.7/26.5
Instance Split (IS) – Supervised Learning Experiments									
Greedy (baseline)	21.7/22.6	8.97/21.5	18.5/21.0	22.7/21.2	10.2/21.1	12.8/21.1	22.3/21.3	20.6/21.3	17.3/21.6
Uniform (baseline)	9.41/20.0	3.65/20.0	7.91/20.0	11.1/20.0	5.01/20.0	3.63/20.0	15.5/20.0	16.7/20.0	8.41/20.0
Swin-T + Emb.	23.1/35.1	33.7/41.0	20.3/28.8	16.7/18.6	17.7/29.5	26.3/34.3	24.5/29.1	17.5/26.5	22.5/30.8
Swin-B + Emb.	22.0/34.0	29.4/36.5	17.7/26.1	16.7/17.0	17.1/30.2	25.0/34.2	26.2/30.7	21.5/29.6	21.6/29.9
Cross-Transformer + Emb.	20.5/30.4	6.3/15.3	15.5/22.9	15.1/15.6	8.7/23.9	10.7/18.2	21.7/24.7	19.0/27.3	14.7/22.8
ViT-16 + Emb.	25.6/36.4	39.7/47.1	21.2/30.8	15.5/16.3	20.1/33.8	39.4/40.8	29.0/33.0	20.3/29.6	25.9/33.5
MAE + Emb.	25.4/36.7	34.2/43.2	21.6/31.5	16.4/16.7	20.0/33.3	32.0/39.7	28.2/32.9	18.6/26.6	24.5/33.0
SimSiam + Emb.	39.9/49.4	8.61/19.2	40.2/49.7	24.1/26.1	14.1/23.2	29.5/39.6	34.3/37.2	37.5/43.7	27.6/35.3
R18 + Emb.	44.0/54.0	8.8/19.8	41.1/47.6	24.5/26.7	13.7/26.5	30.9/40.2	43.3/45.5	29.5/34.8	29.4/37.4
R50 + Emb.	42.1/51.9	8.47/19.6	41.3/47.7	25.1/27.1	12.8/22.6	32.9/42.8	45.6/48.0	41.8/47.9	29.7/37.3
R50 + GloVe	46.0/56.3	39.2/48.5	53.9/56.4	26.7/28.9	21.5/32.4	58.9/68.5	48.5/50.4	43.3/47.8	40.0/47.2
R50 + GPT2	47.0/57.9	44.8/53.1	55.1/58.6	26.1/28.4	27.2/39.3	61.0/71.3	49.0/50.2	42.5/48.4	42.1/49.6
R50 + BERT	48.5/59.3	46.1/54.9	56.7/60.2	26.5/28.4	28.5/39.7	65.6/75.4	44.3/46.2	39.9/45.3	42.8/50.2
CLIP	41.3/52.9	18.2/29.3	33.3/41.1	19.8/21.9	12.9/24.9	27.8/42.8	32.2/36.2	29.9/36.1	27.3/36.4
FLAVA	47.7/58.1	20.2/29.7	41.4/47.1	25.4/27.1	19.6/31.2	30.5/41.9	33.2/35.7	38.3/44.2	32.3/40.2
Answer Split (AS) – Answer Generalization Experiments									
R50 + BERT (FT + Cls.)	0.1/23.8	1.5/13.2	0.0/16.8	0.0/1.6	0.4/17.3	0.0/21.1	0.0/6.0	0.0/15.0	0.19/10.2
R50 + BERT (FT + Reg.)	12.0/28.4	10.4/25.7	19.6/30.8	9.5/10.6	3.64/18.3	9.42/28.6	14.1/21.1	25.5/30.9	16.3/23.4

Table 3. SMART-101 performances of various image and language backbones in our framework on PS, FS, IS, and AS splits. We also report the second-grader performances. Each entry shows $S_{\text{acc}}/O_{\text{acc}}$ (%; higher is better). The image backbones in IS are all fine-tuned.

Method	$S_{\text{acc}} \uparrow$	$O_{\text{acc}} \uparrow$
Instance split		
R50 + BERT	42.8	50.2
No meta learning/MTL	29.7	37.3
Image only (no question)	28.3	36.3
Question only (no image)	15.1	23.2
Single image head	25.0	34.3
Few-shot split		
R50 + BERT	16.7	26.5
No meta learning/MTL	14.7	25.2

Table 4. Ablation studies using the R50 + BERT model.

6. Conclusions

We started by asking the question: *are deep neural networks SMARTer than second graders?* Our analysis in Table 3 shows that the performances of SOTA deep models are significantly below second graders on SMART-101 (77% against 20%). Surprisingly, even under the supervised setting (IS) – when the networks have seen similar instances of a puzzle – the performance is inferior (43%). However, with sufficient training data, SOTA models do demonstrate some level of learning algorithmic skills (e.g., arithmetic, spatial reasoning, *etc*.), yet struggle on simple algebra or path tracing problems. To conclude, the answer to our overarching question is clearly *no*, and there appears to be a significant gap in the perceived competency of AI models and their true algorithmic reasoning abilities. We hope SMART-101 offers a solid step to make advancements in that direction.[6]

Acknowledgements: We thank Joanna Matthiesen (CEO, Math Kangaroo USA) for providing the human performance statistics and permission to use the MK puzzle images in this paper.

[6]More details, experiments, and results are in our extended paper [12].

References

[1] GLPK toolkit. https://www.gnu.org/software/glpk/. 5

[2] NetworkX Python package. https://networkx.org/. 5

[3] Math Kangaroo USA, NFP Inc. https://mathkangaroo.org/mks/, 2012–2022. 2, 3, 4

[4] Somak Aditya, Yezhou Yang, Chitta Baral, and Yiannis Aloimonos. Combining knowledge and reasoning through probabilistic soft logic for image puzzle solving. In *Uncertainty in artificial intelligence*, 2018. 3

[5] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016. 3

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3

[7] David G.T. Barrett, Felix Hill, Adam Santoro, Ari S. Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning*, 2018. 2

[8] Yaniv Benny, Niv Pekar, and Lior Wolf. Scale-localized abstract reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12557–12565, 2021. 2

[9] Yonatan Bitton, Nitzan Bitton Guetta, Ron Yosef, Yuval Elovici, Mohit Bansal, Gabriel Stanovsky, and Roy Schwartz. WinoGAViL: Gamified association benchmark to challenge vision-and-language models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 3

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 7

[12] Anoop Cherian, Kuan-Chuan Peng, Suhas Lohit, Kevin Smith, and Joshua B Tenenbaum. Are deep neural networks SMARTer than second graders? *arXiv preprint arXiv:2212.09993*, 2022. 7, 8

[13] François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019. 1, 3

[14] Claire Cook, Noah D Goodman, and Laura E Schulz. Where science starts: Spontaneous experiments in preschoolers' exploratory play. *Cognition*, 120(3):341–349, 2011. 3

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 7

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[17] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, et al. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32):e2123433119, 2022. 1

[18] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020. 3

[19] Thomas G Evans. *A program for the solution of a class of geometric-analogy intelligence-test questions*. Air Force Cambridge Research Laboratories, Office of Aerospace Research, 1964. 2

[20] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. 1

[21] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 2, 6

[22] Alison Gopnik, Clark Glymour, David M Sobel, Laura E Schulz, Tamar Kushnir, and David Danks. A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3, 2004. 3

[23] Hyowon Gweon, Joshua B Tenenbaum, and Laura E Schulz. Infants consider both the sample and the sampling process in inductive generalization. *Proceedings of the National Academy of Sciences*, 107(20):9066–9071, 2010. 3

[24] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008. 5

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 7

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 7

[27] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 5

[28] José Hernández-Orallo, Fernando Martínez-Plumed, Ute Schmid, Michael Siebers, and David L Dowe. Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, 230:74–107, 2016. 2, 3

[29] Michael Hersche, Mustafa Zeqiri, Luca Benini, Abu Sebastian, and Abbas Rahimi. A neuro-vector-symbolic architecture for solving raven's progressive matrices. *arXiv preprint arXiv:2203.04571*, 2022. 3

[30] Douglas R Hofstadter. *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought.* Basic books, 1995. 2

[31] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *AAAI Conference on Artificial Intelligence*, volume 35, 2021. 2

[32] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[33] Huaizu Jiang, Xiaojian Ma, Weili Nie, Zhiding Yu, Yuke Zhu, and Anima Anandkumar. Bongard-HOI: Benchmarking few-shot visual reasoning for human-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[34] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 2, 3

[35] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Thirty-fourth Conference on Neural Information Processing Systems*, 2020. 3

[36] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. DeepStory: Video story QA by deep embedded memory networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2016–2022, 2017. 3

[37] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1, 3

[38] Richard Lehrer and Leona Schauble. Supporting inquiry about the foundations of evolutionary thinking in the elementary grades. 2012. 3

[39] Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. Adversarial VQA: A new benchmark for evaluating the robustness of VQA models. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE international conference on computer vision*, pages 10012–10022, 2021. 7

[41] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 3

[42] Mikołaj Małkiński and Jacek Mańdziuk. A review of emerging research directions in abstract visual reasoning. *arXiv preprint arXiv:2202.10284*, 2022. 2

[43] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961. 2

[44] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988. 2

[45] Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021. 1

[46] Lauren J Myers and Lynn S Liben. Graphic symbols as "the mind on paper": Links between children's interpretive theory of mind and symbol understanding. *Child Development*, 83(1):186–202, 2012. 3

[47] Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Anima Anandkumar. Bongard-LOGO: A new benchmark for human-level concept learning and reasoning. *Advances in Neural Information Processing Systems*, 33:16468–16480, 2020. 2, 3

[48] OpenAI. Gpt-4 technical report, 2023. 1

[49] Niv Pekar, Yaniv Benny, and Lior Wolf. Generating correct answers for progressive matrices intelligence tests. In *Advances in Neural Information Processing Systems*, 2020. 2

[50] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 7

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 7

[53] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. 7

[54] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[55] Joshua S Rule, Joshua B Tenenbaum, and Steven T Piantadosi. The child as hacker. *Trends in cognitive sciences*, 24(11):900–915, 2020. 1

[56] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[57] Shailaja Keyur Sampat, Yezhou Yang, and Chitta Baral. Visuo-linguistic question answering (VLQA) challenge. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 4606–4616, 2020. 3

[58] Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Alberto Lopez Magana, Wojciech Galuba, Devi Parikh, and Douwe Kiela. Human-adversarial visual question answering. In *Thirty-fifth Conference on Neural Information Processing Systems*, 2021. 3

[59] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019. 2

[60] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 1

[61] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 3, 7

[62] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[63] Atharv Sonwane, Sharad Chitlangia, Tirtharaj Dash, Lovekesh Vig, Gautam Shroff, and Ashwin Srinivasan. Using program synthesis and inductive logic programming to solve bongard problems. *arXiv preprint arXiv:2110.09947*, 2021. 3

[64] Steven Spratley, Krista Ehinger, and Tim Miller. A closer look at generalisation in RAVEN. In *European Conference on Computer Vision*, 2020. 2, 3

[65] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Annual Meeting of the Association for Computational Linguistics*, 2019. 3

[66] Damien Teney, Peng Wang, Jiewei Cao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. V-PROM: A benchmark for visual reasoning using visual progressive matrices. In *AAAI Conference on Artificial Intelligence*, volume 34, 2020. 2

[67] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5228–5238, 2022. 3

[68] Lara M Triona and David Klahr. A new framework for understanding how young children create external representations for puzzles and problems. In *Notational Knowledge*, pages 159–178. Brill, 2007. 3

[69] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. CrossFormer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations, ICLR*, 2022. 7

[70] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment task for visually-grounded language learning. In *Visually Grounded Interaction and Language (ViGIL) Workshop at the Thirty-second Conference on Neural Information Processing Systems*, 2018. 3

[71] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[72] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5317–5327, 2019. 2, 3

[73] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2287–2305, 2019. 3

[74] Wenhe Zhang, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1332–1340, 2020. 3

[75] Tao Zhuo, Qiang Huang, and Mohan Kankanhalli. Unsupervised abstract reasoning for raven's problem matrices. In *IEEE Transactions on Image Processing*, 2021. 2