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
Are Deep Neural Networks SMARTer than Second Graders?

Anoop Cherian\textsuperscript{1} Kuan-Chuan Peng\textsuperscript{1} Suhas Lohit\textsuperscript{1} Kevin A. Smith\textsuperscript{2} Joshua B. Tenenbaum\textsuperscript{2}

\textsuperscript{1}Mitsubishi Electric Research Labs (MERL) \textsuperscript{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\textsuperscript{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.”

\textit{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

\textsuperscript{1}The SMART-101 dataset is publicly available at: \url{https://doi.org/10.5281/zenodo.7761800}

\textsuperscript{2}The answer to the puzzle in Figure 1 is: C.

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 \textit{bird}. The answer is shown below\textsuperscript{2}. 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 \cite{alphago}), language-guided image generation (e.g., the recent DALL-E-2 \cite{dalle2} and ImageGen \cite{imagegen}), creative story writing (e.g., using GPT-3 \cite{gpt3}), solving university level math problems \cite{university_math}, algorithmic inference \cite{algorithmic_inference}, and general question-answering/dialog (e.g., ChatGPT \cite{chatgpt} 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 \cite{introspection, new_tasks, further_work}.

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
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 Bondgard 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 Hofstadter’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
Machine Number Sense [74]

Bongard-LOGO [47]

Bongard-HOI [33]

ARC [13]

Machine Number Sense [74]

RAVEN [72]

Image riddles [4]

VLQA [57]

PororoQA [36]

CLEVR [34]

SMART-101 (ours)

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.

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 $A$, and the task is to have an AI model $f_\theta$, parameterized by $\theta$, that can provide the correct answer $a$ to a given problem tuple $(I, Q, A)$, i.e.,

$$f_\theta(I, Q) \rightarrow a \in A.$$  (1)

To learn the parameters $\theta$ of the model $f_\theta$, we use a dataset $\mathcal{R} = \{\pi_1, \pi_2, \ldots, \pi_K\}$ consisting of a set of $K = 101$ distinct puzzles. We call each $\pi$ 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 $\mathcal{P}_\pi = \{p_{\pi_1}, p_{\pi_2}, \ldots, p_{\pi_n}\}$, where $p_{\pi}$ denotes a new instance of root puzzle $\pi$. Thus, our full dataset $\mathcal{D} = \bigcup_{\pi \in \mathcal{R}} \mathcal{P}_\pi$.

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.

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.
3.2. 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 programatically augment the puzzles via re-making 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

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.
to learn an “algorithm” to solve a puzzle (rather than using only the perception modules), we randomly change the visual, linguistic, 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 include 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 $X = [X_{11}, X_{12}, X_{21}, X_{22}]$, 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 $[N, N(N-1)/2]$. 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

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

3Note that this categorization was done among the authors via a manual categorization and voting on the root puzzles.
Mathematically, let $g_\alpha$ and $\ell_\beta$ 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 $\alpha$ and $\beta$ 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 $h^\gamma_\pi$, to the output of the image backbone, where $h^\gamma_\pi$ is specific to each root puzzle $\pi$ and has its own parameters $\gamma$. Using these modules, our prediction model for puzzle $\pi$ is:

$$f^\theta_\pi(I, Q) := \text{pred}^\gamma_\pi(\text{fuse}_\nu((h^\gamma_\pi(g_\alpha(I)) + \ell_\beta(Q)))),$$  
(2)

where $\text{fuse}_\nu$ denotes a shared MLP to fuse the image and language embeddings and $\text{pred}^\gamma_\pi$ 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 $\text{pred}^\gamma_\pi$ would be an RNN, while for another puzzle, the response could be an integer in 1–100, for which $\text{pred}^\gamma_\pi$ could be an MLP classifier with 100 softmax outputs. We abstractly represent trainable parameters in various modules by $\theta$.

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

$$\min_{\Theta} \mathbb{E}_{\pi \sim \mathcal{R}} \mathbb{E}_{(I, Q, a) \sim \mathcal{P}_\pi} \text{loss}_\pi(f^\theta_\pi(I, Q) - a),$$  
(3)

where $\Theta = \cup_{\pi \in \mathcal{R}} \{\theta\}_\pi$ and $\text{loss}_\pi$ is a puzzle-specific loss that is activated based on the root puzzle $\pi$ 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 $\text{loss}_\pi$ could be: (i) a softmax cross-entropy loss (selecting in a discrete answer range) or (ii) an $\ell_1$ regression loss predicting a scalar value.

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

$$\hat{a} = \arg\max_{\alpha \in \mathcal{A}} \text{sim}_\pi(f^\theta_\pi(I, Q), \alpha),$$  
(4)

where $\text{sim}_\pi$ captures the similarity of a predicted answer value against the choices in $\mathcal{A}$, and $\text{sim}_\pi$ is specific to the problem $\pi$ (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). 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 $S_{\text{acc}}$ that computes the frequency with which the correct solution was produced by a model and (ii) the option selection accuracy $O_{\text{acc}}$ 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

---

4In 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\(^3\) 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 fine-tune 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 $\sim$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 ($\sim$40% on $S_{acc}$ for ViT-16, $\sim$34% for Swin-T and MAE, etc.), while they perform the least on tasks that need path tracing. We find that pretrained 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 use 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 ($\sim$12%) improvement in performance. Our results also confirm that both vision and language are essential to solve SMART-101.

---

\(^3\)All the pretrained backbone models are downloaded from public repositories, specifically https://huggingface.co/models.
### 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.\(^6\)

\(^6\)More details, experiments, and results are in our extended paper [12].
References


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


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

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. Adversarial grades. 2012. 3

Richard Lehrer and Leona Scauble. Supporting inquiry about the foundations of evolutionary thinking in the elementary grades. 2012. 3

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


