A Review of Emerging Research Directions in Abstract Visual Reasoning

Mikołaj Małkiński and Jacek Mańdziuk

Abstract—Abstract Visual Reasoning (AVR) problems are commonly used to approximate human intelligence. They test the ability of applying previously gained knowledge, experience and skills in a completely new setting, which makes them particularly well-suited for this task. Recently, the AVR problems have become popular as a proxy to study machine intelligence, which has led to emergence of new distinct types of problems and multiple benchmark sets. In this work we review this emerging AVR research and propose a taxonomy to categorise the AVR tasks along 5 dimensions: input shapes, hidden rules, target task, cognitive function, and main challenge. The perspective taken in this survey allows to characterise AVR problems with respect to their shared and distinct properties, provides a unified view on the existing approaches for solving AVR tasks, shows how the AVR problems relate to practical applications, and outlines promising directions for future work. One of them refers to the observation that in the machine learning literature different tasks are considered in isolation, which is in the stark contrast with the way the AVR tasks are used to measure human intelligence, where multiple types of problems are combined within a single IQ test.

Index Terms—Abstract Visual Reasoning, Deep Learning

1 INTRODUCTION

Abstract Visual Reasoning (AVR) domain encompasses problems that require formulating analogies between abstract visual concepts instantiated in varying scenarios. Generally, in such tasks the core challenge is to recognise relations that govern simple 2D shapes and their attributes among potentially many images. These problems often-times require the solver to apply previously gained skills, knowledge and experience in a completely new setting and are therefore considered as a well-performing predictor of human intelligence (IQ) [1, 2]. Examples of AVR tasks include the commonly recognized Raven’s Progressive Matrices (RPMs) [3, 4], Odd-one-out (O3) tasks [5, 6] or Bongard Problems (BPs) [7], as well as the emerging ones that focus on extracting and applying visual analogies [8], combine visual and arithmetic reasoning [9], explicitly focus on extrapolation [10], require the ability to differentiate between same and different concepts [11], or test generative modelling in few-shot learning setting [12].

On the one hand, a recent stream of research has revealed that current machine learning (ML) methods, especially those that fall into the deep learning (DL) bucket, can solve certain AVR problems posed in simpler settings exceptionally well [13, 14]. Other works, however, have pointed out that AVR tasks can form a rich and challenging testbed for evaluating different generalisation capabilities of the tested ML/DL methods [15]. Many of these generalisation challenges remain unsolved to date. Furthermore, various AVR benchmarks have been recently proposed (e.g. [8, 9, 16]) that constantly uncover consecutive shortcomings of modern ML methods.

With the increased number of available AVR benchmarks, more and more methods are proposed to tackle them. Even though in many cases the methods operate on similar inputs and outputs, most of the time they are evaluated only on a single chosen task, without considering the

Fig. 1. AVR taxonomy. The discussed AVR tasks can be cataloged along 5 dimensions: 1) the input shapes present in panels of the problem instance; 2) the types of hidden rules that govern shapes and their attributes; 3) the target task being solved; 4) the cognitive function imitated when solving the problem; 5) the main challenge posed by the task.
Fig. 2. AVR and related domains. The field of AVR includes problems that require reasoning about abstract concepts present in the image or identifying hidden rules that govern the visual entities. Although such concepts are also present in related domains of VQA, VCR and PR, the AVR field differs from them mainly by neither utilizing natural language input (as in VQA and VCR) nor referring to the real-world objects (as in VCR and PR).

large set of remaining available problems. As a consequence, highly specialized models that excel in one task may be over-fitted to a particular problem and struggle with even slightly different tasks (e.g. CoPINet [37] was able to solve matrices from RAVEN [38] almost perfectly, but was later shown to achieve much worse results on I-RAVEN [39]).

1.1 Motivation
This review collates existing AVR problems in terms of several dimensions, which can be used by ML practitioners to make an informed decision when selecting a benchmark for evaluating selected generalisation capabilities of the tested approaches.

By pointing out the commonalities and differences of AVR tasks, we invite the AVR community to consider multiple AVR problems at once when working on new approaches. Such a setup better aligns with how AVR tasks are used to measure human intelligence and is still a grand challenge for Artificial Intelligence (AI) and ML systems.

On a general note, research on universal multi-task approaches to solving diverse AVR problems may help understand the operational principles of human intelligence and may be a step towards accomplishing Artificial General Intelligence (AGI).

1.2 Taxonomy
To better describe and explain the links and differences among a variety of AVR tasks, we propose the following taxonomy that allows to explore and catalogue the AVR problems along 5 dimensions (Fig. 1):

1) Input shapes. AVR problems are composed of multiple panels, each of them potentially containing multiple shapes. The objects can be of different forms, including geometric shapes drawn from a limited vocabulary, and abstract shapes that come from a wide set of possibilities and rarely repeat across problems.

2) Hidden rules. Similarly to the input shapes, AVR tasks may comprise explicit rules known beforehand that are sampled from a fixed vocabulary. Such rules most often refer to logical operators, such as OR (disjunction), AND (conjunction), XOR (exclusive disjunction), etc., combined with other well defined concepts, such as progression or constancy. On the other hand, AVR tasks may contain a practically unlimited set of abstract rules defined loosely by means of intuitive visual concepts or expressed in natural language.

3) Target task. AVR problems can be designed with various target tasks in mind. In the vast majority of the cases these are: classification, where the goal is to select an answer from a limited set of choices; generation, where a missing image (or part of it) has to be generated; description, where the answer has to be described in natural language.

4) Cognitive function. We further categorise the AVR problems based on the cognitive function that is required to solve them. This division includes problems that focus on: completion, where the goal is to fill-in the problem matrix with an answer; discrimination, where a rule (or a set of rules) has to be discovered that separates the provided panels.

5) Main challenge. The last perspective we identify is the type of challenge that underlines a given AVR problem. This involves: domain transfer, where a concept learned in a source domain has to be transferred to a target domain; extrapolation, that requires solving problems with novel values of certain attributes, not observed in the learning phase; arithmetic, which tests the ability of reasoning about numbers and arithmetic operations from visual inputs.

1.3 Scope
In a broader perspective, the field of AVR belongs to the Visual Reasoning domain which encompasses tasks that require reasoning, i.e. identifying and extracting task-relevant information from visual input (usually an image or a set of images).

The domains most related to AVR are presented in Fig. 2. All of them address the problem of reasoning about abstract concepts present in the image or identifying hidden rules (also referred to as patterns) that govern visual entities, although in various settings. First of all, such tasks often emerge in the field of Visual Question Answering (VQA) [22–26], where the goal is to answer a question written in natural language referring to an associated image. In VQA, the information present in the image is sufficient for answering the related questions, whereas the Visual Commonsense Reasoning (VCR) field [27–31] takes it a step further and places the tasks in real-world settings, where
external knowledge is often required to solve them. The concept of real-world settings is also utilised in the Physical Reasoning (PR) problems [32–36] that comprise environments governed by underlying physical mechanisms or rules.

In contrast, AVR does not depend on real-world knowledge – in fact, it even assumes that the test-taker will be unfamiliar with the presented shapes and their attributes. Furthermore, AVR matrices are not associated with instance-specific questions, but rather a general short description of the whole problem is provided. In comparison to PR problems where objects are governed by physics, AVR puzzles rely on logical relations or abstract rules defined through visual analogies.

A few existing works have already attempted to review the progress in constructing and solving AVR tasks using classical [17] and more recent [18–20] methods. In [18], the author discusses recent progress in AI systems toward analogy-making and conceptual abstraction, and shows how far the current approaches are from the human level. The work puts into question the role of the popular RPM benchmarks in measuring the analogy-making capacity due to their large training sets and advocates for relying on BP and Abstraction and Reasoning Corpus (ARC) [12] that offer few-shot learning setups, which are more aligned with how analogy-making process is evaluated in humans. A similar perspective is taken in [19], where the authors refer to recent works on RPMs and BPs as demonstrations of how limited current AI systems are — they are able to solve single benchmarks at most and rarely generalise to other problems even from the same domain. The topic of relational reasoning and concept learning is further explored in [20], where the shortcomings of current ML approaches to solving AVR tasks and limitations of selected benchmarks are discussed.

In contrast, in this work we refrain from focusing on exemplary benchmarks and instead provide a comprehensive review of available AVR tasks that were proposed to date, some of them not mentioned in the existing surveys. We align the AVR problems with the proposed taxonomy which should be helpful in making informed decisions about which AVR benchmarks should be selected in future works. While prior works mainly criticise AVR benchmarks due to their misalignment with approaches used for evaluating human analogy-making abilities, this review presents an optimistic view of the field as a whole, describes fundamental challenges that are posed by AVR in the context of computational intelligence, and advocates for their significance in other streams of research.

Our review also differs from similar prior works in the choice of described methods for solving AVR tasks. While [18] discusses progress across 3 fundamental approaches: symbolic methods, probabilistic program induction, and deep learning, the author reviews only a handful of proposed models. Similarly, selected chosen methods are described in [20] and their shortcomings are outlined. The authors additionally hypothesize that attention mechanisms may be used as remediation.

Instead of advocating for a single approach or reviewing selected examples from a wide category of methods, in this survey we concentrate on the most promising recent DL models. This allows to cover a wide spectrum of approaches and provides a unified view of recent strategies for designing DL-based solvers.

1.4 Contribution

This survey describes both leading AVR benchmarks on which the state-of-the-art was already advanced considerably, as well as emerging problems that highlighted the major shortcomings of current pattern analysis algorithms and are yet to be solved. Overall, this paper:

- comprehensively reviews emerging types of AVR problems and benchmarks;
- introduces an AVR taxonomy that aligns existing AVR tasks in terms of their commonalities and differences;
- provides a unified view on current DL approaches to solving AVR problems, that is grounded in the introduced taxonomy;
- links advances in AVR to other domains of practical applicability and presents possible future research paths.

1.5 Structure

We continue the survey by reviewing existing AVR problems and datasets in Section 2. In Section 3 we categorise the outlined AVR tasks according to the proposed taxonomy. Referring to the above categorisation, in Section 4 we review a variety of approaches that were proposed to solve these tasks, with the main focus on DL methods. Section 5 presents a summary discussion of the AVR literature, identifies connections between AVR and other fields, and presents ideas for future work. Section 6 concludes the survey.

2 EMERGING AVR PROBLEMS

Even though there exists a wide landscape of AVR tasks and logic puzzles designed for human solvers [5, 42, 43], e.g. popular Octomino puzzles (Fig. 3), the majority of current DL research in this area is unquestionably devoted to solving RPMs [3, 4, 21]. Recent works, however, start to consider other than RPM formulations of AVR problems that allow
to assess and analyse the capabilities of DL approaches from complementary perspectives.

In general, an AVR problem consists of a set of images that depict simple 2D shapes, which differ in some visual attributes. The images are arranged in a meaningful problem-specific structure that is often realised in some form of a grid. At the heart of these tasks lie the abstract rules that describe the relations between images, the encompassed objects, and their attributes. When solving an AVR task, the main goal is to discover this underlying structure that can later be applied to provide the final answer. The method of providing the answer depends on the problem and can be formulated in several setups such as classification, generation, or description in natural language. In the remainder of this section, we describe contemporary AVR tasks one by one and then align them with the AVR taxonomy in Section 3.

2.1 Raven’s Progressive Matrices

Being the most often discussed AVR task in DL literature, RPMs were chosen as a proxy for studying machine intelligence for a plethora of automatic pattern analysis algorithms. To a large extent, this comes from their wide-spread presence in the cognitive literature, where RPMs were found to be well-suited for testing human intelligence [2] and highly diagnostic of human abstract and relational reasoning abilities [1].

Another key contributor to the number of DL publications that tackle RPMs is a wide suite of their automatic generation methods. Despite a relatively small—from DL perspective—number of instances in the initial set of hand-crafted RPMs [3, 4], automatic methods allow generating a huge number of instances that are required for current data-hungry ML methods. Preliminary works include Sandia matrix generation software [40], synthetic RPMs [41], and G-set – a set of RPMs that was used to train the DeepIQ system [14]. To better understand the shortcomings of current ML approaches in solving RPMs, PGM [15] and RAVEN [38] datasets were introduced that enable measuring performance in specific generalisation regimes and contain images with compositional structure. Apparently, the answer panels in RAVEN dataset were biased, which was discovered and mitigated in subsequent works that introduced I-RAVEN [39] and RAVEN-FAIR [44]. Example matrices from RPM datasets are shown in Fig. 4.

Classical Raven’s matrices consist of two components. The set of context panels contains 8 images arranged in a 3 × 3 grid with a missing bottom-right image. The goal of the test taker is to choose a matching image from the second component – a set of up to 8 answer panels. To select the correct answer, it is required to identify hidden rules (e.g. constancy, progression, logical operators such as AND or XOR) that govern the shapes (e.g. circle, star, triangle) and their attributes (e.g. color, rotation, size) present in the images. Depending on the dataset, the rules may be applied row-wise, column-wise, or—sometimes—also diagonally. After identifying the rules and the way they are applied, an answer that preserves all the rules, after being placed in the bottom-right panel, has to be chosen. In the default setup, realised in all the datasets described in this work, there is only a single answer panel that preserves all the rules, whereas the remaining ones satisfy fewer number of rules or none of them.

RPMs are the subject of a recent survey, where existing benchmarks, DL models and learning methods are reviewed [21]. The authors also discuss how RPMs facilitate development of successful methods that address practical applications. We further explore this topic in Section 5, where we argue that not only RPMs, but also other AVR tasks are relevant and influential for various research streams.
2.2 Visual Analogy Problems

In RPM tasks the solver has to identify abstract patterns that govern the objects and their attributes and select an answer for which the same rules are instantiated. However, the images in an RPM are oftentimes similar – one can say they come from the same domain. On the other hand, the ability to make analogies is best verified when a given rule or concept from a source domain has to be applied to entities from a different target domain [45, 46]. Visual Analogy Problems (VAPs) cover a wide suite of benchmarks where such setting can be formed. In general, VAPs are composed of several panels divided into two parts. The first part (source domain) presents a relation, which has to be reinstated in the second part (target domain), e.g. by completing or inpainting a missing panel. Most common in the literature are VAPs composed of 4 panels, where the final completed matrix should satisfy the following symmetry: A is to B, as C is to D (also written as A:B:C:D). However, emerging AVR benchmarks additionally consider extended problem formulations.

To verify the capabilities of DL models in such analogy making setting, a novel instantiation of VAPs was proposed [8]. VAPs of this type are structurally similar to RPMs, i.e. they contain a set of context images arranged in a 2 × 3 grid with a missing bottom-right panel and the test-taker is supposed to complete the matrix with an appropriate answer panel chosen from a set of 4 candidate images. Exemplary VAPs from [8] are shown in Fig. 5.

The process of making analogies is often difficult, as it requires to encapsulate a concept implemented in a source domain in such an abstract way that would allow its flexible application to different target domains. Remarkably, the authors of the dataset discovered that even simple DL models can be equipped with this ability given appropriate training setup.

To this end, they propose a method of learning analogies by contrasting abstract structure (LABC) [8]. LABC generates the set of candidate panels in a deliberate way, that determines difficulty of the matrix. The method differentiates between two kinds of candidates: perceptually plausible panels – that simply belong to the target domain; semantically plausible panels – that not only match the target domain, but in addition make the bottom row satisfy some relation after being completed with this panel. However, only the correct answer makes the bottom row satisfy the same rule as demonstrated in the source domain. This deliberate way of pairing the correct answer with semantically plausible candidates discourages shortcut solutions – the underlying method is unable to choose the correct answer simply by checking if the candidate matches the target domain (perceptually plausible), or if the bottom row completed with the candidate satisfies any rule (semantically plausible), but also the rule has to match the one presented in the source domain.

The resultant learning process facilitates structural alignment that was shown to be of crucial importance when transferring knowledge between problems [47–49]. The VAPs dataset [8] contains matrices with only perceptually plausible answers, as well as matrices with semantically plausible answers to test how the evaluated methods perform in a more demanding regime.

2.3 Odd-one-out

In both RPMs and VAPs the goal is to complete a set of images with a single matching panel that preserves the hidden rules that govern the objects and their properties. In the odd-one-out (O³) problems, the set of images is already extended with a panel that breaks the pattern – the odd one. O³ problems have been widely studied from both cognitive and computational angles [6, 50–53] and commonly appear in human puzzles [5]. In fact, the challenge of identifying an odd element in a set of objects has been recognised as fundamental not only to humans, but also to other animals [54, 55].

Despite crucial importance, O³ problems have been considered only recently as a benchmark for ML methods. In [14], the authors created two sets—G1-set and S1-set—of O³ matrices using images analogous to RPM datasets – G-set (Fig. 4c) and S-set, respectively. Examples of O³ tasks are shown in Fig. 6. Each problem instance is composed of $n \in \{4, 5, 6\}$ images that depict a single centrally-located object. Each instance contains a subset of $n - 1$ panels, that share up to 3 common properties among {color, rotation, shape, size} and there are no other $n - 1$ panels that would share the same or greater number of features. The odd element is defined as the one that differs in each of these shared properties (see Fig. 6). By employing similar objects...
Odd-one-out problems [14]. Each problem instance contains a set of images, out of which one differentiates from the rest (see description in Section 2.3). Odd elements are marked with a dotted red boundary, with an explanation provided in the problem’s caption. The matrices come from the G1-set [14].

Fig. 6. Odd-one-out problems [14]. Each problem instance contains a set of images, out of which one differentiates from the rest (see description in Section 2.3). Odd elements are marked with a dotted red boundary, with an explanation provided in the problem’s caption. The matrices come from the G1-set [14].

as in G-set and S-set RPM benchmarks, the G1-set and S1-set datasets offer a unique testbed for evaluating transfer learning capacity of the tested methods.

2.4 Bongard Problems

Although some prior works have shown an impressive performance of DL models in solving RPMs [21, 56], it was also identified that such models may struggle when the availability of data is scarce [57]. Moreover, recent literature reviews have criticized existing RPM benchmarks for their large sets of training instances, which casts doubt on the broader impact of the reported improvements in solving RPMs [18, 20]. On the other hand, Bongard Problems (BPs) [7] are composed of abstract shapes and rules, which allow forming a rich testbed for evaluating learning methods in a low-sample regime.

The initial BP matrices [7] were hand-crafted and therefore only few hundred problems were manually constructed by individual contributors [58]. BP examples are shown in Fig. 7. Each problem instance is composed of two parts, each of them containing 6 images arranged in a grid. Each part is governed by a distinct abstract rule. The test-taker has to discover these rules that differentiate both parts. Contrary to the already discussed problems, the rules in BPs are not sampled from a fixed, pre-defined set and additionally have to be described (by the test-taker/solving system) in natural language.

There have been two more approaches that framed the task of solving BPs to make it somehow attainable for the current learning systems. In [59] the initial set of hand-crafted BPs [7] is considered. Each problem is framed as a binary classification task. First, one image is extracted from each matrix part, and then for each of them, the model has to point out the part from which the image was extracted. This contrasts with the problem formulation proposed in the seminal work [7], where matrices are solved by describing the separating rules in natural language.

A similar problem formulation is proposed in [16], where the Bongard-LOGO benchmark is introduced. The dataset measures the human-level concept learning and reasoning of AI agents with the help of 12000 matrices. Contrary to the approaches from [59, 60], for each problem, the test set is not extracted from the context matrices, but two additional test images are associated with each problem instance (see Figs. 7c–7f). Moreover, Bongard-LOGO defines four testing splits that help to evaluate different generalization capabilities of the tested methods in a similar fashion to PGM regimes [15] in the RPM task. The abstract shapes presented in matrices from Bongard-LOGO are randomly generated by executing sequences of program instructions written in action-oriented LOGO language [61].

Similarly to RAVEN, performance evaluation of human test-takers was conducted on Bongard-LOGO and compared to several meta-learning [62, 63] and DL methods [64–66]. The DL approaches demonstrated, to some extent, the ability to solve the BP matrices in a simpler problem setting, where the description of rules in natural language is replaced by a binary classification task. However, a study on human solvers conducted in [16], revealed that even current state-of-the-art meta-learning methods lag far behind the human concept-learning performance.

2.5 Arithmetic Visual Reasoning

The majority of abstract visual reasoning tasks require the solver to reason about abstract patterns composed of primitive objects (e.g. squares, triangles, stars, etc.) with certain attributes (size, rotation, location, fill-in pattern, etc.). However, cognitive human studies have shown that there is another aspect of human intelligence that naturally emerges from vision, namely the sense of numbers [69], that refers to the understanding of numbers and associated operations, as well as the ability to solve the related mathematical (usually arithmetic) problems. The sense of numbers was identified as a fundamental component of early human development [70, 71] and the interpretation of a magnitude expressed in a symbolic form was found to be highly indicative for human competency in solving diverse mathematical tasks [72].

Although prior works have shown that neural models can be applied to mathematical reasoning problems expressed in the form of text [73–75] or symbolic [76] inputs, a recent work [9] has demonstrated substantial limitations of current DL approaches when simple arithmetic problems are integrated into AVR puzzles. The authors introduced the Machine Number Sense (MNS) dataset [9] that evaluates the capability of intelligent systems to understand numerical symbols with relational operations between them (crystallised intelligence), and in adaptive problem-solving (fluid intelligence). Each MNS problem is composed of 3 images of simple geometric figures. The problems may come in 3 different types: combination integrates few geometric shapes into a specific spatial configuration – in problems with 2
shapes, the structures overlap (Fig. 8a) or one is included in the other (Fig. 8c), while in problems with 3 shapes the structures are tangent to each other (Fig. 8b); composition (Fig. 8d) arranges a set of small geometric shapes into a larger structure (a line, a cross, a triangle, a square, or a circle); partition (Figs. 8e and 8f) divides a single geometric shape into several parts. The geometric shapes used in these problems are similar to the already discussed AVR tasks and include a triangle, a square, a circle, a hexagon, and a rectangle. In contrast to other AVR problems, visual arithmetic matrices additionally contain integers from 1 to 99 located in certain pre-defined locations. For a given problem instance, the test-taker is supposed to discover an arithmetic expression satisfied by the printed numbers and apply it to compute a missing number in the rightmost matrix panel. The expression may use 4 fundamental operators: addition, subtraction, multiplication, and division.

To measure the ability of DL models to solve such arithmetic visual problems, several baseline DL architectures were tested including CNN [77], LSTM [78], RN [32], and ResNet [79]. All tested models struggled with solving the provided matrices and achieved results much worse than a group of human solvers – on average, humans were able to solve 78% of matrices from MNS, while the best performing DL model managed to tackle only 25% of them. Even though current DL approaches have demonstrated some ability to solve arithmetic tasks [73–76], the above-described study [9] revealed that problems located at the intersection of mathematics and AVR remain generally challenging for ML/DL methods.

### 2.6 Visual Extrapolation Problems

One of the reasons why current DL approaches still lag far behind humans in various AVR problems is their weak ability to extrapolate, which some works have argued is a key characteristic of human intelligence [80]. The outcomes that support this thesis were emphasised in the context of solving RPMs, where DL models performed poorly in the extrapolation regime of PGM dataset [15, 81]. Similar observations were made in VAPs from [8] where, in general, the DL
methods achieve weaker performance when extrapolating concepts to novel domains.

To measure the extrapolation capabilities of DL methods, Webb et al. [10] proposed two benchmark sets: the Visual Analogy Extrapolation Challenge (VAEC) and the dynamic object prediction task (DOPT) illustrated in Figs. 9 and 10, respectively. Both datasets contain problems that test various extrapolation realizations in terms of scale and translation.

Specifically, matrices from VAEC present yet another instantiation of VAPs, which are oriented towards extrapolation - given images $x_1$, $x_2$ and $x_3$, a missing image $x_4$ has to be selected from a set of up to 6 possible answers (in this example there are 4 choices, A-D) such that the relation between $x_3$ and $x_4$ is analogous to the relation between $x_1$ and $x_2$. Given 4 possible axis of variation (brightness, size, x coordinate, y coordinate), the relation between $x_1$ and $x_2$, as well as between $x_3$ and $x_4$ varies in the same direction and with the same magnitude. The set of answers contains only such choices that vary along the same axis, however, only one image correctly completes the analogy. This approach of defining available answer panels resembles the LABC method [8] where the set of answers contains only semantically plausible candidates. However, in contrast to VAPs from [8], matrices from VAEC explicitly focus on extrapolation in terms of brightness, scale and translation, while the benchmark proposed by Hill et al. [8] focuses on

2. Denoting L=left, R=right, T=top, B=bottom, C=center, the remaining answers are:
(b) $L + T + R = 11 \implies ? = 1$
(d) $TL - BR = TR - BL = TC - BC = RC - LC = 16 \implies ? = 47$
(f) $\Delta - \nabla = \Delta - \nabla = \Delta - \nabla = 9 \implies ? = 30$

Fig. 8. Visual arithmetic problems [9]. In order to solve these tasks, the test-taker has to identify an arithmetic expression presented in the first 2 panels and use it to find the value of a missing number in the rightmost panel. The answers can be deduced as follows: (a) in the left and center image, dividing left number by the right number gives 15, so $15/? = 15 \implies ? = 1$; (c) in both outer and inner ring, the following holds: $left \times top/right \times bottom = 77$, so $21 \times 1/21 \times ? = 77 \implies ? = 77$. (e) this involves performing the following arithmetic operation: $(left - top) \times right + bottom$, which for both left and center image gives 77, so $(20 - 16) \times ? + 21 = 77 \implies ? = 14$; To demonstrate the difficulty of these matrices, the answers to examples (b), (d) and (f) are hidden and the readers are encouraged to attempt solving them. The answers are provided in the footnote.

Fig. 9. Visual analogy extrapolation problem [10]. Given images $x_1$, $x_2$, $x_3$, the goal is to select the image $x_4$ from a set of up to 6 possible answers (in this example there are 4 choices, A-D) such that the relation between $x_3$ and $x_4$ is analogous to the relation between $x_1$ and $x_2$. Given 4 possible axis of variation (brightness, size, x coordinate, y coordinate), the relation between $x_1$ and $x_2$, as well as between $x_3$ and $x_4$ varies in the same direction and with the same magnitude. The set of answers contains only such choices that vary along the same axis, however, only one image correctly completes the analogy. This approach of defining available answer panels resembles the LABC method [8] where the set of answers contains only semantically plausible candidates. However, in contrast to VAPs from [8], matrices from VAEC explicitly focus on extrapolation in terms of brightness, scale and translation, while the benchmark proposed by Hill et al. [8] focuses on
In many real-world scenarios, one is often faced with a same-different task. Given a sequence of images, the goal is to recreate the $T'$th (rightmost) image that follows the pattern. To test extrapolation capabilities, the training images contain only small squares with width $w \in [3, 13]$ (measured in pixels), whereas the test images contain large squares with $w \in [13, 31]$. The panels are of size $64 \times 64$ pixels.

Fig. 10. Dynamic object prediction task [10]. Given a sequence of $T - 1$ images of squares, the goal is to recreate the $T'$th (rightmost) image that follows the pattern. To test extrapolation capabilities, the training images contain only small squares with width $w \in [3, 13]$ (measured in pixels), whereas the test images contain large squares with $w \in [13, 31]$. The panels are of size $64 \times 64$ pixels.

Forming analogies between different domains.

VAEC matrices fall into 2 regimes that test different types of extrapolation: invariance to translation in terms of size, brightness and location; and invariance to scale between visual objects. Problems in each regime are divided beforehand into train and test splits, which allows measuring generalisation of the tested methods to values from a much bigger range than encountered during the learning phase.

In addition, the authors proposed DOPT, a complementary extrapolation task. Given a sequence of images $x_t$, $\ldots$, $x_T-1$ that depict a gradually changing and moving square observed at $T = 1$ consecutive time-steps, the task is to predict square’s final size and position at time $T$ (see Fig. 10). To form an extrapolation benchmark, train matrices depict only small squares with width $w \in [3, 13]$, while test matrices contain larger squares with $w \in [13, 31]$.

The authors of [10] have shown that traditional DL approaches fail to demonstrate consistent performance in any of the two regimes.

To facilitate the extrapolation ability of neural models, the Temporal Context Normalization (TCN) [10] method was proposed, which relies on a task-relevant temporal window over which the representations are normalised. This approach allows neglecting absolute feature magnitudes while preserving the in-between feature relations. The proposed normalisation technique excelled in both considered extrapolation tasks and additionally improved results of Hill et al. [8] on the VAP benchmark discussed in section 2.2.

2.7 Same-different tasks

In many real-world scenarios, one is often faced with a question whether a given object is same or different from another object. This task is so fundamental for living beings that such discriminatory ability emerged in a variety of animal species [82–84]. The difficulty of this task rises when the differences are subtle, or when the concept of sameness concerns abstract properties of the compared objects. In same-different (S-D) tasks from AVR domain, objects that are same shouldn’t be interpreted based on pixel-to-pixel similarity, but rather based on abstract visual concepts they present. Consequently, different objects not only differ in terms of individual pixels, but additionally present distinct—sometimes even opposite—abstract visual concepts. Visual reasoning models for solving the S-D tasks received well-deserved attention in recent works [85, 86].

A seminal work [11] has shown that differentiating between same and different concepts has proven difficult to learn by the current ML models. Namely, Fleuret et al. [11] proposed the Synthetic Visual Reasoning Test (SVRT) which comprises 23 types of classification problems. Each problem type defines an abstract rule that differentiates images from 2 categories (left and right matrix parts). The abstract rules are built on concepts such as symmetry, relative position, proximity, and a few more, with examples presented in Fig. 11.

Given a problem type, a matrix is created by sampling images from both categories. The number of sampled images determines the matrix difficulty – the more images the matrix contains, the easier the abstract rule should be to discover. Next, given a test image, initially not assigned to any of the 2 categories, the test-taker has to select the category to which the test image belongs. The images present irregular randomly-generated black and white closed contours with no additional distracting attributes, which allows unbiased evaluation of human (or machine) reasoning skills irrespective of previous experience.

Preliminary attempts to tackle the SVRT challenge with DL systems demonstrated that traditional CNN architectures struggle with solving the S-D tasks [87, 88]. However, a parallel work on a related problem revealed that such
unsatisfactory results may arise not from the weak expressive power of DL models, but rather from optimization difficulties and naive learning curriculum [89]. By injecting task-relevant information into intermediate-level representations, Gülçehre and Bengio [89] demonstrated that a seemingly weak model may achieve close to perfect results on these tasks, given appropriate supervision.

Motivated by these findings, subsequent works further explored the topic. In [90, 91], the authors analysed in detail the successes and failures of CNN architectures on the SVRT dataset. It was discovered that the models handled categorization problems concerning the relative spatial position of shapes relatively well, while at the same time struggled with problems that involve rotation, translation, reflection, and identification of symmetries. To facilitate a more fine-grained analysis of the problem, the authors introduced the parametric SVRT dataset (PSVRT [90, 91]), which remains an open challenge for DL systems until today.

On the other hand, even though it was concluded in [90] that DL models struggle with solving S-D tasks, it was later shown in [92–94] that the initial categorization problems from SVRT can be solved with ResNets – modern DL models that excel in computer vision.

Alternatively, in [95] the authors show how a specific image pre-processing technique can make even CNN-based classifiers competitive in solving matrices from SVRT.

### 2.8 Abstraction and Reasoning Corpus

Though the described AVR benchmarks pose a real challenge for contemporary learning systems on their own, there is a fundamental difference in how these tests are used in measuring machine intelligence versus how they are used in psychometric tests for humans. Namely, current benchmarks consist of huge training sets, that not only allow the trained systems to prepare in advance for the tasks they will encounter during the evaluation phase, but more crucially are known beforehand by the developers of these models. As a result, current learning systems are hand-crafted for a particular task they will be evaluated on – a topic we further explore in Sections 4 and 5. This critical difference between human and machine evaluation settings sparked a series of discussions in the AVR community [18] and motivated the development of the Abstraction and Reasoning Corpus (ARC), with examples shown in Fig. 12.

The dataset shifts the evaluation setting from test problems that are known in advance, into an uncharted territory more aligned with human psychometric tests where the types of tasks found in the training and test sets do not intersect.

More specifically, ARC comprises 1000 distinct AVR tasks divided into training (400 instances) and test (600 instances) sets. Each task presents a unique challenge illustrated with several examples. In each task, the goal is to understand the problem we are faced with, based on the analysis of the exemplary demonstrations, so as a task-specific test example can be solved. While the method of solving the task is unique in each case, the tasks share a similar structure – each instance contains an input and an output grid, where each cell is coloured with one from a fixed set of 10 possibilities. Grids can be of variable size ranging from $1 \times 1$ up to $30 \times 30$, and the test output grid with an answer has to be created from scratch by the test-taker.

Compared to the previously described AVR tasks, ARC doesn’t divide the problem instances into train and test sets based on specific object attributes, but rather by the task being solved. Each task encountered in the testing phase is novel for the solver, which recreates a setting found in human psychometric tests and limits the amount of task-specific experience – the specific skills required for solving the presented task can only be gathered from few demonstrations.

Crucially, tasks from ARC do not focus on measuring how well can the tested method learn particular skills, but rather measure how efficient the learning system is in acquiring new skills from merely a few examples.

### 3 Fusion of AVR tasks

To better understand the commonalities and differences among emerging AVR research tasks and the respective
TABLE 1
Alignment of tasks with AVR taxonomy. AVR problems and their respective benchmarks are catalogued into the following 5 dimensions of the AVR taxonomy: Input shapes, Hidden rules, Target task, Cognitive function, Main challenge.

| Problem                          | Dataset          | Geometric shapes | Abstract shapes | Explicit rules | Abstract rules | Classification | Generation | Description | Completion | Discrimination | Domain transfer | Extrapolation | Arithmetic |
|----------------------------------|------------------|------------------|-----------------|----------------|----------------|----------------|-------------|-------------|------------|----------------|----------------|---------------|------------|
| Raven’s Progressive Matrices     | Sandia [40]      | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          | ✓              | ✓              | ✓             | ✓          |
|                                  | Synthetic [41]   |                  |                 |                |                |                |             |             |            |                |                |               |            |
|                                  | G-set [14]       | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
|                                  | RAVEN [38, 39, 44]| ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
|                                  | PGM [15]         | ✓                | ✓               |                |                |                |             |             |            |                |                |               |            |
| VAPs                             | Hill et al. [8]  | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Bangard problems                 | Hand-crafted [96]| ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
|                                  | Bangard-LOGO [16]| ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Same-different                   | SVRT [11]        | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Odd-one-out                      | G1-set [14]      | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
|                                  | S1-set [14]      | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Visual arithmetic reasoning       | MNS [9]          | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Extrapolation                    | VAEC [10]        | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
|                                  | DOPT [10]        | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |
| Abstraction and reasoning         | ARC [12]         | ✓                | ✓               | ✓              | ✓              | ✓              | ✓           | ✓           | ✓          |                |                | ✓             | ✓          |

3.1 Input shapes
As presented in Figs. 3-12, AVR problems are composed of multiple panels, and each of them may comprise various shapes (see Fig. 13). On a general note, we propose to differentiate between two types of shapes: geometric and abstract. This distinction allows us to gain a clear perspective on: 1) solving which problems requires the ability to adapt to ever-changing inputs (abstract shapes); 2) for which tasks knowledge of the finite vocabulary of shapes is enough to solve them (geometric shapes).

3.1.1 Geometric shapes
The prevalent number of AVR problems refer to matrices instantiated from a fixed set of geometric shapes that are rather easy to recognise and comprehend. This genre of problems includes Raven’s Progressive Matrices, Visual Analogy Problems, Odd-one-out tasks, Visual Arithmetic Reasoning matrices, and Extrapolation problems. Moreover, these shapes often appear in various variants and are characterised by certain properties. For instance, the objects presented in Sandia matrices are defined by: 1) a shape: oval, rectangle, diamond, triangle, trapezoid, the letter T; 2) a variant: wide, narrow, tall, short; 3) an attribute: shading, orientation, size, count. While such characteristics allows producing perceptually rich matrices, a solver with access to a finite vocabulary is generally able to fully comprehend the presented objects.

3.1.2 Abstract shapes
On the contrary, certain AVR tasks are composed of abstract shapes that come from a wide set of possibilities and rarely repeat across problems. Such shapes underlie Bongard Problems and S-D tasks. The use of a unique selection benchmarks, we further align them along the 5 dimensions of the introduced AVR taxonomy. An aggregated summary is presented in Table 1 and each dimension is discussed in detail in the following sections. Table 2 presents references to the publicly available datasets.

TABLE 2
Public datasets. Majority of the discussed datasets are publicly available and can be found at the addresses listed below. Datasets marked with GitHub are available on GitHub and can be accessed at the following address: https://github.com/{address}.

| Dataset                                | Address                                      |
|----------------------------------------|----------------------------------------------|
| Sandia [40]                            | LauraMatzen/Matrices                         |
| G-set [14]                             | deepiq/deepiq                               |
| RAVEN [38]                             | WellyZhang/RAVEN                             |
| I-RAVEN [39]                           | husheng12345/SRAN                           |
| RAVEN-FAIR [44]                        | yanyibenny/RAVEN_FAIR                       |
| PGM [15]                               | deepmind/abstract-reasoning-matrices         |
| Hill et al. [8]                        | deepmind/abstract-reasoning-matrices         |
| Hand-crafted [96]                      | foundalis.com/res/bps/bpidx.htm              |
| Bangard-LOGO [16]                      | NVLabs/Bangard-LOGO                         |
| SVRT [11]                              | fleuret.org/git/svrt                        |
| G1-set [14]                            | deeppiq/deepiq                              |
| S1-set [14]                            |                                              |
| MNS [9]                                | zwh1999anne/Machine-Number-Sense-Dataset    |
| VAEC [10]                              | taylorwwebb/learning_representations_that_support_extrapolation |
| DOPT [10]                              |                                              |
| ARC [12]                               | fchollet/ARC                                |
of atypical shapes leads to several consequences. Firstly, it allows comparing human and machine visual reasoning performance in a fair setup, where both parties encounter the shapes for the first time. Secondly, such a representation removes the need for complex perceptual modules that have to recognize various aspects of real-world images, such as texture or occlusion, that are often crucial for an accurate understanding of the scene.

Consequently, this setting allows to fully focus on comprehension of the abstract relations present in the AVR tasks, and hence is especially well-suited for evaluating AVR capabilities. We believe that abstract input setting should be a preferred mode in future AVR research.

3.2 Hidden rules

Even though AVR tasks with abstract shapes require the solving agent to be adaptable to novel inputs, the resultant AVR matrices, anyway, contain rather simple input shapes. However, this perceptual simplicity may be often misleading, as the main difficulty of AVR problems lies in their hidden rules [97], which is further exacerbated if multiple rules are applied [2]. Similar to the input shapes, AVR tasks can be divided into two main categories, depending on the type of the underlying rules. We will refer to these categories as explicit and abstract, respectively. Table 3 presents typical examples.

3.2.1 Explicit rules

In the majority of AVR problems the rules come from a fixed and explicit set of well-defined possibilities. Such explicit rules are often joint with geometric shapes. Surprisingly, modern AVR benchmarks with explicit rules contain only a handful of unique relations, e.g. PGM contains 5 different rules: progression, XOR, OR, AND, consistent union; whereas RAVEN contains 4 such rules: constancy, progression, arithmetic, distribute three. Even though the set of available rules is limited to few elements only, its coupling with different shapes and attributes allows to generate huge suites of problem instances (e.g. PGM contains more than 1 million RPMs in each of its 8 regimes). Quite surprisingly, AVR tasks generated from such seemingly small sets of rules often pose a challenge even for human solvers [38].

3.2.2 Abstract rules

Contrary to AVR benchmarks with explicit rules, their counterparts with abstract relations are usually much more diverse. Although abstract rules are often defined in less rigorous ways, they turned out to be really useful in determining the abstract reasoning skills of the test-taker. As an example, both Bongard problems and same-different tasks employ abstract rules that define differences between two groups of panels. This perspective places the solver in a never-seen-before scenario, which can be approached only by applying previously gained skills and knowledge in an entirely new setting. Such problems lie at the heart of human-level intelligence [46] and are more aligned with human IQ tests. For the above reasons, we advocate for considering such AVR problem setups as a more accurate approximation of human-level intelligence in machines.

3.3 Target task

In general, when solving AVR problems, the main difficulty for humans lies in discovering the abstract structure that governs the matrix. After gaining this understanding, humans can solve AVR tasks irrespectively of whether the solution should be provided by selecting an answer from a fixed set of choices (classification), recreating an element that matches the sequence (generation), or expressing the answer in natural language (description). In contrast, pattern recognition algorithms often struggle with solving certain target tasks, while excelling at others.

3.3.1 Classification

Classification—being the most often studied target task in the contemporary AVR literature—requires the solver to select an appropriate panel from the set of possible answers. This task is offered in nearly all existing benchmarks, with the exception of the hand-crafted set of BPs and the matrices from DOPT. In RPMs and VAPs, the test-taker is given a pre-defined set of possible answers, out of which only one correctly completes the matrix. In BPs from Bongard-LOGO, the goal is to assign two test images to matching categories shown in left and right problem parts. A similar approach is taken in the SVRT benchmark, which requires assigning a test image to one out of two categories. Matrices from MNS also fall into this category, as the solver has to complete the matrix with an appropriate answer selected from a pre-defined set of choices, which in the case of MNS are integers ranging from 1 to 99.

Moreover, some problems offer an auxiliary target task consisting in recognition of additional meta-data related to a
given RPM task, the learning system may additionally predict the hidden rules that govern the matrix \[15, 38, 81\].

3.3.2 Generation

Alternatively, instead of explicitly selecting an answer that correctly completes the matrix from a set of choices, a generative model may be considered that recreates the missing image (or part of it). Such a target task lies at the core of problems in DOPT and is further considered in the context of RPMs, where a generative model first recreates the whole missing panel and then selects the most similar one from the set of pre-defined choices \[98, 99\].

3.3.3 Description

The idea behind the initial set of hand-crafted BPs was to create a set of problems that represent abstract concepts which have to be discovered and described in natural language. However, due to a limited number of problem instances, this formulation of a target task hasn’t been considered to date in any of the proposed solutions. While image captioning is a flourishing field with visible progress in recent years \[100–102\], none of the existing methods tackles the problem of describing the abstract concepts present in AVR tasks in natural language. Besides the mentioned set of hand-crafted BPs, there aren’t any other benchmarks that could be used for evaluating the quality of image captioning methods in AVR settings.

Definitely, developing such benchmark sets along with suitable methods for solving them remains an open problem and a challenging research path for future work.

3.4 Cognitive function

The entire domain of AVR problems is a rich testbed for two fundamental cognitive functions: completion and discrimination, which are highly relevant components of perceptual learning \[103–105\]. In general, a successful AVR problem solver has to excel in both.

3.4.1 Completion

At the heart of many discussed AVR tasks lies the mechanism of completion, where a selected sequence of images has to be completed such that all underlying patterns are preserved. This genre of tasks includes RPMs, VAPs, matrices from MNS and both extrapolation settings from VAEC and DOPT. Completion is often coupled with the target task of classification where an element that completes the sequence has to be chosen from a fixed set of provided answers, or sometimes with generation where the missing element has to be recreated from scratch.

3.4.2 Discrimination

A complementary cognitive function, that is required in practically all but MNS and DOPT problem instances, is centred around discrimination between multiple possible answers. For instance, when solving RPMs, a natural approach is to attempt to complete the matrix with each of the answer panels and then select the best-fitted one by means of their direct comparison. The ability to discriminate is also required in BPs, O\(^2\), and S-D tasks, where a set of hidden rules that separate the provided panels has to be discovered. Many works in cognitive literature have suggested that learning by discriminating positive from negative examples is of critical importance in the human learning process \[106–108\] and additionally promotes structural alignment \[48\] that is beneficial in knowledge transfer \[47, 49\].

3.5 Main challenge

AVR tasks pose a unique challenge on their own. As illustrated by the already described dimensions of the taxonomy, these problems allow analysing capabilities of reasoning models from various perspectives. Although, at first glance, they may seem like a set of puzzles with little relevance to other domains and more practical settings, in fact, quite the opposite is true. AVR problems are often built around much more general challenges and the methods that excel in these tasks are likely to be revolutionary in other areas, as well. Before we explore these connections in Section 5, we first introduce the main challenges posed by AVR problems.

3.5.1 Domain transfer and knowledge generalisation

In contemporary literature, AVR problems and related benchmark sets are often used to measure domain transfer abilities of the tested methods in multiple ways: (1) RAVEN contains matrices belonging to 7 different structural configurations, which can be used to verify whether the tested method is able to understand matrices with shapes arranged in novel ways \[38, 109\]; (2) PGM defines 8 regimes that test model’s ability to generalise to novel types of problem instances, e.g. to matrices with the same rules applied to novel objects and attributes, or to matrices with rules never encountered during the learning phase; (3) VAPs from Hill et al. \[8\] explicitly define problem instances where the solver has to identify the rule that governs the objects in a source domain and apply it to the objects in a (different) target domain. Another example is DeepIQ \[14\], a model trained to solve RPMs from the G-set, able to utilise this knowledge for solving Sandia RPMs and odd-one-out tasks.

Despite certain efforts, knowledge transfer between AVR problems remains a highly underexplored topic and to date no satisfactory solution has been proposed for this problem setting.

3.5.2 Extrapolation

A problem directly related to domain transfer is extrapolation, where a model trained to operate on a fixed set of values of certain attributes is evaluated on problems with novel values of these attributes. Such distinct generalisation regimes appear in: (1) PGM, where train matrices contain shapes with lighter colour and smaller size, while test instances comprise of shapes with darker colour and bigger size; (2) VAPs from Hill et al. \[8\], which borrow the concepts from PGM’s extrapolation regime and additionally extend the number of sides a shape can have and the quantity of these shapes; (3) Bongard-LOGO, where test matrices contain shape patterns generated by action programs with one additional instruction compared to the training ones, resulting in shapes with larger diversity.

What is more, both VAEC and DOPT directly focus on evaluating extrapolation by presenting sequences of rectangles with changing position and size. Each such sequence
However, the majority of existing learning systems are many tasks reaches or even exceeds human level [14, 56]. This is largely attributed to their impressive performance that in the majority of recent literature gravitates toward DL systems. This progress made in these seminal works is comprehensively summarised in [17].

The progress made in these seminal works is comprehensively summarised in [17]. The progress made in these seminal works is comprehensively summarised in [17]. The progress made in these seminal works is comprehensively summarised in [17]. The progress made in these seminal works is comprehensively summarised in [17].

Another distinct challenge posed by AVR tasks resides at the intersection of visual and arithmetic reasoning. The topic of solving math problems with computational methods has its roots in early domain literature [110–112] and has received high interest in recent ML works that consider problems with symbolic [113–115] or text [73, 75, 116] inputs. Merely, endowing ML models with arithmetic reasoning turned out to be a staggering challenge in non-trivial problem settings [75]. However, only recently has the MNS dataset been extended with an additional image that follows the pattern. The key difficulty lies in the data distribution – train sequences contain rectangles of smaller size and model evaluation is performed on sequences of larger rectangles.

### 3.5.3 Arithmetic reasoning

Another distinct challenge posed by AVR tasks resides at the intersection of visual and arithmetic reasoning. The topic of solving math problems with computational methods has its roots in early domain literature [110–112] and has received high interest in recent ML works that consider problems with symbolic [113–115] or text [73, 75, 116] inputs. Merely, endowing ML models with arithmetic reasoning turned out to be a staggering challenge in non-trivial problem settings [75]. However, only recently has the MNS dataset been extended with an additional image that follows the pattern. The key difficulty lies in the data distribution – train sequences contain rectangles of smaller size and model evaluation is performed on sequences of larger rectangles.

So far, no automatic learning method has achieved satisfactory performance, and the domain has yet to wait for effective machine solutions.

### 4 A unified perspective on AVR models

Inspired by the importance of AVR tasks in the evaluation of human intelligence, many works embarked on the quest to measure the performance of learning systems in solving these problems. Initial approaches included models that employed hand-crafted rules [96, 117], structure-mapping theory [118–121], or similarity-based methods [122–126]. The progress made in these seminal works is comprehensively summarised in [17].

Despite promising results of the first above-mentioned computational models for solving AVR problems, the majority of recent literature gravitates toward DL systems. This is largely attributed to their impressive performance that in many tasks reaches or even exceeds human level [14, 56]. However, the majority of existing learning systems are geared towards solving a single chosen task and are rarely evaluated in other settings. Consequently, many models are discussed in isolation, despite clear similarities between the approaches. In what follows we present a unified perspective on recent DL models for solving diverse AVR tasks. We hope that this fusion of various computational threads from the recent AVR literature will be helpful in clearing the path towards general learning systems (Artificial General Intelligence) capable of solving disparate AVR problems.

In the existing DL systems for solving AVR tasks, three general components can be identified: 1) encoder: that defines the way of processing and representing input panels of the problem instance; 2) fusion: which determines how image representations are aggregated to fuse information from different matrix parts; 3) decoder: specifying how the model arrives at its answer based on the latent problem representation.

While in some cases it is not clear where the boundaries between these components should be precisely set, this rough partitioning allows gaining a clear understanding of what do different models have in common and where do they differ. The above unified view on AVR models is depicted in Fig. 14.

#### 4.1 Encoder

In general, each AVR problem instance can be considered as a set of panels (images) $X = \{x_i\}_{i=1}^n$, where $n$ is a dataset specific number of separate images in each problem instance (e.g. $n = 16$ for RPMs from Sandia shown in Fig. 4a and $n = 4$ for O$^3$ tests from G1-set shown in Fig. 6a).

Encoder, being the first component in each DL-based model for solving AVR tasks, defines how a raw panel (image) $x_i$ is converted to a hidden representation $h_i$. $h_i$ is later aggregated with embeddings $\{h_j\}_{j \neq i}$ of other matrix panels in the fusion layer, so it is crucial that $h_i$ preserves as much semantic context as possible about the specific image.
The most common encoding approach is to use a simple neural block that extracts image low-level features. For this purpose, some approaches flatten the image to a 1D sequence of pixels and employ a multi-layer perceptron (MLP) to transform it into a latent representation \( h \in \mathbb{R}^d \), where \( d \) is the dimension of the representation vector [14]. Instead of MLP, other works apply CNNs and then flatten their output to again arrive at \( h \in \mathbb{R}^d \) [36]. Alternatively, an identity transformation may be applied so that the original image is passed as-is to the fusion layer. This idea is being followed in some hierarchical models [39, 57] that are described in the following section.

### 4.2 Fusion

The next fundamental component of AVR models merges representations of individual panels \( \{h_i\}_{i=1}^n \) into a unified representation \( g \). The main goal of the fusion component is to reason about rules that span multiple matrix panels, in contrast to the encoder that extracts panel-specific features.

On the one hand, in the majority of AVR works, the fusion module consists of simple MLPs — composed of linear layers with non-linear activation functions and normalisations. Examples include BatchNorm [127], LayerNorm [128], or Temporal Context Normalisation [10] that proved effective in solving extrapolation tasks.

On the other hand, despite certain similarities and shared components, the way of fusing individual panel representations, actually, differentiates the AVR models most. By injecting diverse inductive biases into the model’s architecture, the models are constructed with a particular problem in mind. Consequently, the initial assumptions about the target task exert a big impact on the range of applicability of these methods to AVR problems.

In what follows, we catalogue existing AVR approaches with respect to the most common inductive biases they are built on.

#### 4.2.1 Separate panels

Common to all contemporary approaches is the fundamental way of representing AVR matrices. In general, when AVR tasks are used in human psychometric tests, the matrix is provided as a single image. Instinctively, humans start solving the matrix by decomposing it into separate panels. In contrast, current AVR benchmarks provide already decomposed matrices as sets of individual panels. Consequently, all existing works rely on this assumption and the availability of already separated representations of individual images forms a basis for the majority of remaining inductive biases.

A common way of exploiting this pre-processed representation is to stack all the images on top of each other and pass the resultant 3D matrix through CNN or ResNet [79]. This approach was evaluated across many AVR tasks [8, 9, 13, 15, 38, 39, 44] and although it usually does not lead to achieving state-of-the-art results, it forms a reliable baseline.

Despite wide popularity, the assumption that an AVR task is provided as a set of separate images is perhaps a major obstacle on the way towards universal systems capable of solving diverse AVR tasks that may come in a variety of forms.

#### 4.2.2 Pre-defined hierarchies

Thanks to the availability of a matrix pre-segmented into individual panels, many works incorporated a problem-specific arrangement of panels directly into the model’s architecture. Moreover, the underlying structure of AVR tasks is usually well-defined, which allows combining these individual panels in a task-relevant manner. For instance, hidden rules in PGM matrices are only applied row-wise, which gives a clear hint into how representations of individual panels should be fused.

Based on these observations several task-specific architectures were proposed: SRAN [39], designed for solving RPMs, gradually aggregates the information from individual images, rows, and pairs of rows (see Fig. 15); CoPINet [37] forms panel representations by adding together features along rows and columns; LEN [129] considers triples of objects only along rows and columns as inputs to a Relation Network [32]; DCNet [57] explicitly creates representations of matrix rows and columns.

While all of these inductive biases turned out to be helpful in solving RPMs, the applicability of the resultant models to AVR tasks with other structures is limited. The introduction of pre-defined hierarchies into model architectures is a common approach for improving results for a particular task at the cost of limiting the applicability to other problems. To make the models suitable for solving diverse AVR tasks, pre-defined panel hierarchies could be replaced with automatic methods for discovering task-relevant panel combinations [130].

#### 4.2.3 Pair-wise relations

Another way of efficient processing of AVR matrices, enabled by the segmentation into individual panels, is to analyse relations between pairs of panel representations. By
comparing pairs of panels a model may, for instance, be trained to efficiently identify differences between them [14].

Most notably, a pair-oriented approach was taken in the WReN model [15, 131] that employs a Relation Network (RN) [32] depicted in Fig. 16. RN operates on a set of objects. In the case of WReN, each object corresponds to a CNN embedding of an individual panel concatenated with the panel’s absolute position in the matrix, expressed as a one-hot vector. The embeddings are concatenated into pairs, each pair is processed by the same MLP, outputs are added together and finally processed by another MLP.

In contrast to the pre-defined hierarchies that are problem-specific, WReN—that considers pair-wise relations—was proven widely applicable to various AVR tasks [8, 9, 15, 16, 38, 39, 44]. Related pair-wise modules [132–134] were found to be of crucial importance in other domains, as well. Specifically, attention-based models can operate on a set of objects represented as a weighted sum of input features that do not necessarily have to correspond to separate input panels. Generally speaking, in contrast to modules with pre-defined hierarchies, modules that consider pair-wise relations are more flexible in solving various tasks.

4.2.4 Single-choice tasks

Further examples of how matrix pre-segmentation into individual panels could be exploited can be observed in the contemporary classification models for single-choice AVR tasks. A typical design choice made in a wide suite of AVR models is to consider the set of matrix panels \( \{x_i\}_{i=1}^n \) (also referred to as the context panels) completed by each of the answer panels \( \{a_j\}_{j=1}^m \) (see Fig. 17). Given the resultant set of panels \( \{x_i\}_{i=1}^n \cup \{a_j\} \), the model can measure how aligned is the considered answer \( a_j \) with the matrix context and express this alignment as a scalar score \( s_j \). With the help of the softmax function \( \sigma \), the array of scores may be transformed into a probability distribution \( \tilde{p} = \sigma([s_1, \ldots, s_m]) \) and the answer corresponding to the highest probability will be chosen as the model’s answer: \( \hat{y} = \arg\max_j \{\tilde{p}_j\}_{j=1}^m \).

This approach is taken in the Wild-ResNet model [15] that stacks the resultant \( n+1 \) panels on top of each other and passes the matrix through ResNet to obtain a single scalar. Wild-ResNet was applied to solve AVR tasks from several benchmarks, including RPMs [15, 38, 39, 44] and VAPs [8].

A related approach is taken in [16] to solve BPs from Bongard-LOGO, where images from the left part together with a test image are stacked on top of each other and passed through a ResNet. Analogous operation is applied to images from the right part. Based on these two representations a discriminatory module produces a score that determines which side the test image aligns with. A similar scheme is employed in WReN, however, in contrast to Wild ResNet, WReN generates an embedding for each of the panels separately. Having the individual embeddings, WReN performs \( m \) parallel passes through the RN module, each time using embeddings of all context panels together with the embedding of a selected answer, which produces a single scalar for the considered answer. Likewise, a WReN-based module is considered in [16] that processes each image independently and then merges their representations using an RN.

While this paradigm of completing the context with one of the answers and generating an alignment score was proven useful in many other works [56, 109, 130, 135–137], the above approaches are fundamentally limited to these single-choice tasks in which the matrix is pre-segmented in advance. Going forward, it will be beneficial to add explicit segmentation modules [138] capable of dividing the matrix into individual panels.

4.3 Decoder

The wide suite of emerging AVR problems poses a number of distinct prediction tasks. In the proposed AVR taxonomy, we have roughly divided them into the following three categories: classification, generation and description. Each of these target tasks requires a unique decoding module capable of producing answers in the expected format. In what follows, we summarise the fundamental approaches in all 3 classes and discuss the importance of the auxiliary tasks that are orthogonal to this categorisation.

4.3.1 Classification

The set of AVR classification tasks mainly includes the already discussed single-choice tests where the context matrix has to be completed with a missing panel (Fig. 17).
In addition, the target task of classification is fundamental to $O^3$ matrices, where an odd element has to be identified amongst the set of matrix panels. To this end, in [14] a heuristics-based scoring module was proposed that analysed feature-based differences extracted by a neural network. In contrast to already described approaches where an alignment score is computed for the matrix completed by one of the answers, the approach proposed in [14] takes a holistic view and considers the matrix as a whole.

Another unique type of a classification task is presented by matrices from MNS where a single integer value has to be predicted that correctly fits into the hidden expression. In this case, the missing integer may be one of 99 available choices and computing an alignment score of each one of them is impractical. As a result, the existing models to solve visual arithmetic problems include a decoder implemented as a linear layer with 99 output neurons and the softmax activation function.

### 4.3.2 Generation

Another type of prediction challenge, posed by some AVR tasks, is image generation [139–141]. Instead of selecting an answer from a pre-defined set of choices, the answer has to be recreated from scratch. Though the task is mainly emphasised in DOPT and ARC benchmarks, some works [98, 99, 142, 143] considered the problem of generating RPM answers. Answer generation enables a supplementary way of interpreting model’s predictions – by looking at what the model considers as an answer, we can roughly determine whether the model recognises the underlying structures or rather merely relies on visual shortcuts.

A common way of approaching an AVR generation task is to construct an autoencoder [144–146] that firstly reduces the input image into a condensed latent representation and then uses this representation to recreate the image. To excel in this task, the latent representation has to contain semantically meaningful information about the image, that can additionally be helpful in the target task of classification.

In some approaches, the task of image generation is treated as an auxiliary task used just for pre-training the encoder [13, 14, 131, 147], while in other works the decoder is trained in parallel with the classification branch [10, 148]. The decoder may be implemented as a sequence of MLPs [14], or as a stack of transposed convolutions [10, 13, 131, 147], in both cases with the sigmoid activation in the last layer.

### 4.3.3 Description

While the spectrum of available classification and generation tasks in the AVR literature is rather wide, tasks involving description didn’t get as much attention. Since the seminal work [7] that introduced Bongard Problems where hidden rules have to be described in natural language, none of the subsequent approaches succeeded in this task. Instead, the problem was rephrased into simpler settings where description is replaced with binary classification [16, 59] (cf. Section 2.4).

On the other hand, recent progress in image captioning [100–102] and natural language generation coupled with scene understanding [149–151] suggests that current learning systems are, in principle, capable of generating descriptions in natural language to reasoning problems with visual input. This, in turn, suggests that the lack of successful methods for describing answers to AVR tasks in natural language may arise not from the lack of capacity of the proposed models, but rather from the unavailability of appropriate datasets on which such models could be trained.

An attempt to combine abstract reasoning with natural language was taken in [152], where the authors enrich ARC with natural language descriptions prepared by human annotators. The descriptions provide instructions about how the matrices could be solved, which resembles the fundamental challenge posed by BPs, where a description of abstract rules that differentiate left and right matrix parts has to be generated.

Still, further research placed at the intersection of AVR and natural language generation is needed to advance the current ML systems in this human-specific ability.

### 4.3.4 Auxiliary tasks

In addition to the already described target tasks, some works consider auxiliary training objectives that are helpful in training the models for a downstream task. Such additional objectives are achieved by attaching another decoder head in parallel to the main decoder, as depicted in Fig. 14. While some already mentioned works consider the image generation task as a supplementary objective, another common approach is to train the model to predict the underlying rules [15]. To this end, the task’s structure is encoded as a multi-hot [15, 38] or one-hot [81] vector and an auxiliary classification head is attached to the model with sigmoid activation in the last layer. This allows predicting possibly many rules that characterise the considered matrix.

Another group of notable auxiliary tasks adapt the family of contrastive losses [153, 154] into the AVR setting, which improves the model’s ability to discriminate between correct and wrong answers [81, 148].

In summary, auxiliary tasks provide yet another means for specifying inductive biases helpful in the considered task, that can bring the model’s focus to the important features identified with expert knowledge. In contrast to inductive biases that are embedded directly in model’s architecture, auxiliary objectives are taken into account only during training, which makes the models easier to adapt to novel tasks.

### 5 Discussion

The AVR domain has seen increasing research interest in recent years. As a result, a wide suite of benchmarks emerged and many methods were proposed to tackle them. At the same time, this increased attention sparked a discussion whether the field moves in the right direction. Some speculations have been made about the unfairness of comparing progress in AVR with human performance. In other discussions, the purpose of training models explicitly to solve AVR tasks is put into question.

While we generally share the above concerns, we believe that even though the progress in AVR domain is not striking, it, nevertheless, step-by-step narrows the gap between human and machine abilities in this highly challenging area.
In what follows, we provide an optimistic perspective on the field and formulate several promising prospects for future research.

5.1 Humans vs Machines
Achieving, or even surpassing, human performance in diverse tasks is the long-standing goal of AI research. However, with the setup of current AVR benchmarks, it is far from obvious how, or even if, a fair comparison between humans and ML approaches can be conducted. In this perspective, some recent works discussed how AVR tasks could be made more aligned with human psychometric tests.

5.1.1 Few-shot vs multiple-epoch training
One of the main differentiators in measuring human and machine performance in solving AVR tasks is the training regime. While humans use AVR tasks mainly for the purpose of IQ evaluation, without extensive learning of how to solve them, current DL systems have to process huge training corpora before achieving any abstract reasoning abilities.

In [12] the author concludes that such training scheme measures how well does the algorithm learn certain skills, while we should be interested in evaluating the system’s ability in acquiring new skills. Similarly, [18] postulates measuring system’s abstract reasoning ability in a few-shot learning setting, which so far no method succeeded in.

While the above observations and claims are definitely valid, we would argue that AVR problems, where domain transfer is the main challenge, still allow making a fair comparison between human and machine performance. In addition to ARC that explicitly tackles these issues by providing a set of matrices with unique tasks, let us take the PGM dataset as an example. It is clear that generalisation regimes other than neutral test the system’s ability to solve novel tasks (please refer to [15] for a detailed description of PGM regimes). In such settings, the model’s performance can’t be improved with additional training examples as shown by the failure of all existing systems in solving matrices from more demanding PGM regimes [21]. Though benchmarks with small training corpora are a good way of ensuring that a fair comparison is made, datasets that test generalisation to novel types of problems, not encountered during training, offer a supplementary surface for comparing human and machine abstract reasoning abilities.

5.1.2 White-box vs black/grey-box solutions
Besides the training scheme, another important issue for conducting a fair comparison is the nature of the considered tasks. A common issue encountered in the evaluation of DL-based systems arises from their lack of interpretability. In effect, some systems with seemingly impressive performance are often found to exploit shortcuts in the datasets which renders the evaluation of their abstract reasoning ability difficult [39]. As pointed out in [18], generative tasks form one way of gaining more confidence that the learning system indeed grasps the task of interest instead of relying on shortcuts that are way easier to miss in classification tasks. In addition, tasks that focus on description form yet another set where interpretation of the model’s predictions could be verified more convincingly.

Referring to the taxonomy proposed in Section 1 it seems fair to say that the results in tasks of generation and description are more trustworthy. In contrast, classification settings might require additional interpretation of how the model arrives at its solution to validate whether the system really understands the task rather than relies on shortcuts.

5.1.3 General vs task-specific approaches
While there are also other crucial points to consider when comparing human and machine reasoning performance [12, 18, 94], the final one most related to this survey is the range of applicability of the learning systems. Current methods are mainly evaluated on particular benchmarks, while humans are known for their intrinsic ability to solve disparate tasks. In order to move forward, multi-task settings need to be considered which better align with the measurements of human intelligence. While constructing general learning systems capable of solving diverse tasks is the grand and yet distant challenge, a first step would be to consider similar tasks according to the dimensions identified in the AVR taxonomy. For instance, designing methods for solving various tasks with familiar geometric shapes, or related explicit rules, might be an achievable intermediate goal.

In summary, we postulate shifting the focus of the community towards solving multiple, but perhaps at first similar, AVR tasks.

5.2 Beyond AVR domain
Despite certain criticism discussed in the previous section, recent progress that originated in the AVR domain has already made an impact in various related areas.

- Discrimination, being one of the cognitive functions fundamental to solving AVR problems, is often found in many practical applications such as robotics [155] or anomaly detection [156, 157]. Contrastive AVR mechanisms [37, 81] that induce discrimination capabilities can be of importance in such tasks.
- The most popular model with roots in AVR is the Relation Network, already applied to a wide array of tasks, including 3D human pose estimation [158], semantic segmentation [159], action recognition [160], reinforcement learning [161], or self-supervised learning of visual representations [162].
- The perspective taken in the O* problem was proven useful for defining a general weakly-supervised framework for learning representations [163].
- Visual Analogy Problems that focus on extrapolation facilitated the construction of a general normalisation technique that can be used not only in AVR, but also in other contexts [10].
- In the spirit similar to the seminal multitask learning paper [164], AVR benchmark sets demonstrated the importance of auxiliary tasks for building more transferable representations [15, 81].
- Also, a recent work [137] used the PGM dataset to evaluate generalisation capabilities of a general compositional model.
On a general note, the main challenges posed by AVR tasks identified in this survey, such as domain transfer, extrapolation, and arithmetic reasoning extend beyond AVR domain and are indispensable for developing human-level AI (or Artificial General Intelligence – AGI). AVR tasks already offer a wide suite of testbeds where initial attempts to develop particular skills of AGI agents can be made.

We are convinced that even if the performance of learning systems in AVR domain is still far from that of humans, or even if the performance gap between humans and machines cannot be assessed with current benchmarks, the underlying challenges posed by existing AVR datasets are of particular importance on the path to accomplishing human-level intelligence.

### 5.3 AVR-like tasks for representation learning

In addition to forming grounds for various generalisation challenges, AVR tasks serve as an inspiration for recent representation learning research, as illustrated in Fig. 18. In [165], the authors propose the V-PROM dataset – a collection of matrices with RPM structure where simple 2D shapes are replaced with real-life images (see Fig. 18a). The matrices present fundamental rules such as counting or logical operators applied to pictures sampled from Visual Genome [166]. The benchmark was created as a testbed for abstract high-level concept learning over real-life objects, which could not be measured with previous AVR datasets.

A new set of VAPs was also constructed in [168]. In contrast to V-PROM, matrices from [168] focus on renderings of realistic cars from the ShapeNet dataset [167], as shown in Fig. 18b. The images vary in texture, shading and viewpoint. Using this dataset, the authors presented a case where a general segmentation model performed better than task-specific architectures trained solely on the automotive VAPs [168].

Yet another perspective was taken in [170], where a self-supervised framework for video representation learning grounded in the O^3 problem is proposed. In this task, sequences of consecutive video frames are considered. From each sequence, a few subsequences are extracted and one of them is perturbed by modifying the temporal order of frames. As presented in Fig. 18c, the goal in this task is to identify such a perturbed (odd) subsequence. Importantly, the training examples can be constructed automatically without human supervision and the authors have shown that a network pre-trained on this self-supervised task learns meaningful representations of videos that are applicable to other downstream tasks such as action recognition on UCF101 [169] and HMDB51 [171] datasets.

While both V-PROM and the dataset with car analogies were constructed as diagnostic tests, in the spirit of [170] another possible application of both these sets could be their utilization as pre-training tasks for representation learning. By focusing on high-level abstract concepts, the tasks might help inducing high-level general representations in visual models, which sets a promising avenue for future work.

Moreover, other datasets conceptually similar to V-PROM and automotive data, pertaining to other real-world domains are highly desirable in order to accelerate research at the intersection of AVR and practical settings.

### 5.4 The main paths to move forward

AVR domain has received a particular renewed interest in recent years, justified by rapid progress in constructing DL
models that excel in solving some of the tasks. Based on these advances, a few promising research directions can be identified.

5.4.1 AVR benchmarks as self-contained challenges

The AVR domain is rooted in the initial attempts of systematic evaluation of human intelligence. While the settings of existing AVR benchmarks are far from those used in human studies, solving AVR tasks in their current form remains a research path of fundamental importance. As already discussed, the AVR domain presents a ready-to-use set of problems that test generalisation abilities still not seen in current learning systems: flexibility, by means of adaptability to novel domains, extrapolation, or visual arithmetic reasoning.

Further research on such methods, facilitated by the current benchmarks is one of the fundamental AVR research prospects.

5.4.2 Towards Artificial General Intelligence

Even though systems that excel in single AVR tasks are of high importance on their own, another path worth exploring is to consider AVR tasks with settings that correspond to human psychometric tests. This way we can not only construct methods capable of possessing certain generalisation abilities, but additionally progress towards more human-like AI (AGI).

Despite impressive developments in recent years and attempts of reproducing human-like approaches [172, 173], existing learning systems are nowhere near human performance [80, 174], and some even claim such levels are unattainable [175]. While no clear path towards AGI seems to exist, the first step is to define how such progress can even be measured. To this end, AVR tasks such as ARC [12] are crucial for measuring the machine progress towards truly intelligent systems.

Certainly, a lot of work is still ahead before the field would truly approach human-level performance in human-like AVR settings.

5.4.3 Machine AI perspective in Human IQ research

While ML research that assumes the above-mentioned alignment of ML experimental conditions with those used in human psychometric tests is undoubtedly one of the grand research avenues in AVR, there is another promising perspective for utilising ML systems that does not require such an alignment.

Recently, several attempts have been made to understand the organizational principles in human visual cortex by analysing how hierarchical DL models operate [176–178]. The researchers discovered that the intermediate representations learned by deep hierarchical CNNs correlate surprisingly well with processes found in biological brains [177, 178], which brought new insights into how a poorly understood area of the visual cortex behaves [176].

Per analogy, in parallel to bringing the human perspective into machine AVR studies, it might be a good time to investigate an opposite viewpoint, by bringing the perspective of machines into the human world. This way we can gain another outlook for investigating human intelligence – just as deep hierarchical CNNs facilitated the formulation of novel insights about the visual cortex, analysis of representations learned by successful AVR machine solvers might help us unravel the mystery of impressive abstract visual reasoning skills in humans.

6 Conclusion

Abstract Visual Reasoning is a flourishing field with rapid progress in recent years. To facilitate future well-informed research, in this work we have surveyed the emerging directions in AVR domain.

We have started with an introduction of the AVR taxonomy, which catalogues existing benchmark sets along 5 dimensions: input shapes, hidden rules, target task, cognitive function, and main challenge. Next, based on this categorisation, we have reviewed the contemporary AVR problems.

Moreover, we have formulated a unified perspective on the introduced DL models for solving AVR tasks. This view allows to distinguish general components relevant in many other settings from problem-specific modules crafted with a particular dataset in mind.

Finally, we have discussed how a fair comparison between human and machine performance can be conducted, shed light on the bright side of AVR research, and discussed the connections between AVR and the representation learning literature.

Going forward, we have advocated for the role of AVR tasks in facilitating research on methods that exert specific generalisation abilities, as well as in accelerating advances towards human-level AI.

References

[1] R. E. Snow, P. C. Kyllonen, and B. Marshalek, “The topography of ability and learning correlations,” Advances in the psychology of human intelligence, vol. 2, no. S 47, p. 103, 1984. 1, 4
[2] P. A. Carpenter, M. A. Just, and P. Shell, “What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test,” Psychological review, vol. 97, no. 3, p. 404, 1990. 1, 4, 12
[3] J. C. Raven, “Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive,” Master’s thesis, University of London, 1936. 1, 3, 4
[4] J. C. Raven and J. H. Court, Raven’s progressive matrices and vocabulary scales. Oxford psychologists Press Oxford, England, 1998. 1, 3, 4
[5] M. Gardner and D. Richards, The colossal book of short puzzles and problems. Norton, 2006. 1, 3, 5
[6] P. E. Ruiz, “Building and solving odd-one-out classification problems: A systematic approach,” Intelligence, vol. 39, no. 5, pp. 342–350, 2011. 1, 5
[7] M. M. Bongard, “The recognition problem,” Foreign Technology Div Wright-Patterson AFB Ohio, Tech. Rep., 1968. 1, 6, 7, 17
[8] F. Hill, A. Santoro, D. Barrett, A. Morcos, and T. Lillicrap, “Learning to Make Analogies by Contrasting Abstract Relational Structure,” in International Conference on Learning Representations (ICLR), 2019. 1, 5, 7, 8, 9, 11, 13, 15, 16
2019, pp. 1075–1087. 2, 15, 18

[38] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu, “Raven: A dataset for relational and analogical visual reasoning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5317–5327. 2, 4, 11, 12, 13, 15, 16, 17

[39] S. Hu, Y. Ma, X. Liu, Y. Wei, and S. Bai, “Stratified rule-aware network for abstract visual reasoning,” in AAAI Conference on Artificial Intelligence (AAAI), 2021. 2, 4, 11, 15, 16, 18

[40] L. E. Matzen, Z. O. Benz, K. R. Dixon, J. Posey, J. K. Kroger, and A. E. Speed, “Recreating raven’s: Software for systematically generating large numbers of raven-like matrix problems with normed properties,” Behavior research methods, vol. 42, no. 2, pp. 525–541, 2010. 4, 11

[41] K. Wang and Z. Su, “Automatic generation of raven’s progressive matrices,” in Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015. 4, 11

[42] M. Gardner, D. J. Albers, and P. L. Renz, Martin Gardner’s mathematical games: the entire collection of his scientific American columns. Mathematical Association of America, 2005. 3

[43] H. E. Dudeney, 536 Puzzles and curious problems. Courier Dover Publications, 2016. 3

[44] Y. Benny, N. Pekar, and L. Wolf, “Scale-localized abstract reasoning,” arXiv preprint arXiv:2009.09405, 2020. 4, 11, 15, 16

[45] D. Gentner, “Structure-mapping: A theoretical framework for analogy,” Cognitive science, vol. 7, no. 2, pp. 155–170, 1983. 5

[46] D. R. Hofstadter, Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought. Basic books, 1995. 5, 12

[47] R. Catrambone and K. J. Holyoak, “Overcoming contextual limitations on problem-solving transfer,” Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 15, no. 6, p. 1147, 1989. 5, 13

[48] D. Gentner and V. Gunn, “Structural alignment facilitates the noticing of differences,” Memory & cognition, vol. 29, no. 4, pp. 565–577, 2001. 13

[49] M. Bassok, “Analogical transfer in problem solving,” The psychology of problem solving, pp. 343–369, 2003. 5, 13

[50] S. Dehaene, V. Izard, P. Pica, and E. Spelke, “Core knowledge of geometry in an amazonian indigene group,” Science, vol. 311, no. 5759, pp. 381–384, 2006. 5

[51] E. S. Gollin and M. Schadler, “Relational learning and transfer by young children,” Journal of Experimental Child Psychology, vol. 14, no. 2, pp. 219–232, 1972.

[52] A. Lovett, K. Lockwood, and K. Forbus, “Modeling cross-cultural performance on the visual oddity task,” in International Conference on Spatial Cognition. Springer, 2008, pp. 378–393.

[53] A. Lovett and K. Forbus, “Cultural commonalities and differences in spatial problem-solving: A computational analysis,” Cognition, vol. 121, no. 2, pp. 281–287, 2011. 5

[54] T. Zentall, D. Hogan, and J. Holder, “Comparison of two oddity tasks with pigeons,” Learning and Motivation, vol. 5, no. 1, pp. 106–117, 1974. 5

[55] T. R. Zentall, D. E. Hogan, and C. A. Edwards, “Oddity learning in the pigeon: Effect of negative instances, correction, and number of incorrect alternatives,” Animal Learning & Behavior, vol. 8, no. 4, pp. 621–629, 1980. 5

[56] Y. Wu, H. Dong, R. Grosse, and J. Ba, “The Scattering Compositional Learner: Discovering Objects, Attributes, Relationships in Analogical Reasoning,” arXiv preprint arXiv:2007.04212, 2020. 6, 14, 15, 16

[57] T. Zhuo and M. Kankanhalli, “Effective abstract reasoning with dual-contrast network,” in International Conference on Learning Representations (ICLR), 2021. 6, 15

[58] H. E. Foundalis, “Index of bongard problems,” http://www.foundalis.com/res/bps/bpidx.htm, 2006, accessed: 2021-07-04. 6

[59] S. Kharagorgiev, “Solving bongard problems with deep learning,” https://k10v.github.io/2018/02/25/Solving-Bongard-problems-with-deep-learning/, Feb. 2018, accessed: 2021-07-04. 6, 17

[60] X. Yun, T. A. Bohn, and C. X. Ling, “A deeper look at bongard problems.” in Canadian Conference on AI, 2020, pp. 528–539. 6

[61] H. Abelson, N. Goodman, and L. Rudolph, “Logo manual,” 1974. 6

[62] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in International Conference on Learning Representations (ICLR), 2016. 6

[63] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in International conference on machine learning. PMLR, 2016, pp. 1842–1850. 6

[64] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Advances in Neural Information Processing Systems, vol. 30, 2017. 6

[65] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, “A simple neural attentive meta-learner,” in International Conference on Learning Representations (ICLR), 2018.

[66] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, “Meta-learning with differentiable convex optimization,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10 657–10 665.

[67] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, “Rapid learning or feature reuse? towards understanding the effectiveness of maml,” in International Conference on Learning Representations, 2020.

[68] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, “A new meta-learner for few-shot learning,” arXiv preprint arXiv:2003.04390, 2020. 6

[69] S. Dehaene, The number sense: How the mind creates mathematics. OUP USA, 2011. 6

[70] K. Wynn, “Addition and subtraction by human infants,” Nature, vol. 358, no. 6389, pp. 749–750, 1992. 6

[71] E. Temple and M. I. Posner, “Brain mechanisms of quantity are similar in 5-year-old children and adults,” Proceedings of the National Academy of Sciences, vol. 95, no. 13, pp. 7836–7841, 1998. 6

[72] M. Schneider, K. Beeres, L. Coban, S. Merz, S. Su-
san Schmidt, J. Stricker, and B. De Smedt, “Associations of non-symbolic and symbolic numerical magnitude processing with mathematical competence: A meta-analysis,” Developmental science, vol. 20, no. 3, p. e12372, 2017.

[73] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay, “Learning to automatically solve algebra word problems,” in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 271–281. 6, 7, 14

[74] D. Huang, S. Shi, C.-Y. Lin, J. Yin, and W.-Y. Ma, “How well do computers solve math word problems? large-scale dataset construction and evaluation,” in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 887–896.

[75] D. Saxton, E. Grefenstette, F. Hill, and P. Kohli, “Analysing mathematical reasoning abilities of neural models,” in International Conference on Learning Representations (ICLR), 2019. 6, 14

[76] G. Lample and F. o. Charton, “Deep learning for symbolic mathematics,” in International Conference on Learning Representations (ICLR), 2019. 6, 7

[77] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in Advances in neural information processing systems, 1990, pp. 396–404. 7

[78] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997. 7

[79] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. 7, 15

[80] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” Behavioral and brain sciences, vol. 40, 2017. 7, 20

[81] M. Małkiński and J. Mańdziuk, “Multi-label contrastive learning for abstract visual reasoning,” 2020. 7, 13, 17, 18

[82] I. M. Pepperberg, “Acquisition of the same/different concept by an african grey parrot (psittacus erithacus): Learning with respect to categories of color, shape, and material,” Animal Learning & Behavior, vol. 15, no. 4, pp. 423–432, 1987. 9

[83] D. L. Oden, R. K. Thompson, and D. Premack, “ Infant chimpanzees spontaneously perceive both concrete and abstract same/different relations,” Child development, vol. 61, no. 3, pp. 621–631, 1990.

[84] A. A. Wright and J. S. Katz, “Mechanisms of same/different concept learning in primates and avians,” Behavioural processes, vol. 72, no. 3, pp. 234–254, 2006. 9

[85] M. Ricci, R. Cadene, and T. Serre, “Same-different conceptualization: a machine vision perspective,” Current Opinion in Behavioral Sciences, vol. 37, pp. 47–55, 2021. 9

[86] K. D. Forbus and A. Lovett, “Same/different in visual reasoning,” Current Opinion in Behavioral Sciences, vol. 37, pp. 63–68, 2021. 9

[87] K. Ellis, A. Solar-Lezama, and J. Tenenbaum, “Unsupervised learning by program synthesis,” in Advances in Neural Information Processing Systems, vol. 28, 2015. 9

[88] S. Stabinger, A. Rodríguez-Sánchez, and J. Piater, “25 years of cnns: Can we compare to human abstraction capabilities?” in International conference on artificial neural networks. Springer, 2016, pp. 380–387. 9

[89] B. Gülçehre and Y. Bengio, “Knowledge matters: Importance of prior information for optimization,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 226–257, 2016. 10

[90] M. Ricci, J. Kim, and T. Serre, “Same-different problems strain convolutional neural networks,” arXiv preprint arXiv:1802.03390, 2018. 10

[91] J. Kim, M. Ricci, and T. Serre, “Not-so-clevr: learning same–different relations strains feedforward neural networks,” Interface focus, vol. 8, no. 4, p. 20180011, 2018. 10

[92] N. Messina, G. Amato, F. Carrara, F. Falchi, and C. Gennaro, “Testing deep neural networks on the same-different task,” in 2019 International Conference on Content-Based Multimedia Indexing (CBMI). IEEE, 2019, pp. 1–6. 10

[93] N. Messina, G. Amato, F. Carrara, C. Gennaro, and F. Falchi, “Solving the same-different task with convolutional neural networks,” Pattern Recognition Letters, vol. 143, pp. 75–80, 2021.

[94] C. M. Funke, J. Borowski, K. Stosio, W. Brendel, T. S. Wallis, and M. Bethge, “Five points to check when comparing visual perception in humans and machines,” Journal of Vision, vol. 21, no. 3, pp. 16–16, 2021. 10, 18

[95] T. Bohn, Y. Hu, and C. X. Ling, “Few-shot abstract visual reasoning with spectral features,” arXiv preprint arXiv:1910.01833, 2019. 10

[96] H. E. Foundalis, “Phaeaco: A cognitive architecture inspired by bongard’s problems.” PhD dissertation, Indiana University, 2006. 11, 14

[97] R. Primi, “Complexity of geometric inductive reasoning tasks: Contribution to the understanding of fluid intelligence,” Intelligence, vol. 30, no. 1, pp. 41–70, 2001. 12

[98] N. Pekar, Y. Benny, and L. Wolf, “Generating correct answers for progressive matrices intelligence tests,” in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 7390–7400. 13, 17

[99] C. Zhang, B. Jia, S.-C. Zhu, and Y. Zhu, “Abstract spatial-temporal reasoning via probabilistic abduction and execution,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9736–9746. 13, 17

[100] S. Bai and S. An, “A survey on automatic image caption generation,” Neurocomputing, vol. 311, pp. 291–304, 2018. 13, 17

[101] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A comprehensive survey of deep learning for image captioning,” ACM Computing Surveys (CSUR), vol. 51, no. 6, pp. 1–36, 2019.

[102] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli,
gence, vol. 34, no. 07, 2020, pp. 12071–12078. 19

[166] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma et al., “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” International journal of computer vision, vol. 123, no. 1, pp. 32–73, 2017. 19

[167] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su et al., “Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015. 19

[168] N. Ichien, Q. Liu, S. Fu, K. J. Holyoak, A. Yuille, and H. Lu, “Visual analogy: Deep learning versus compositional models," arXiv preprint arXiv:2105.07065, 2021. 19

[169] K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012. 19

[170] B. Fernando, H. Bilen, E. Gavves, and S. Gould, “Self-supervised video representation learning with odd-one-out networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3636–3645. 19

[171] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition," in 2011 International conference on computer vision. IEEE, 2011, pp. 2556–2563. 19

[172] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, “Neuroscience-inspired artificial intelligence," Neuron, vol. 95, no. 2, pp. 245–258, 2017. 20

[173] A. Lieto, M. Bhatt, A. Oltramari, and D. Vernon, “The role of cognitive architectures in general artificial intelligence," pp. 1–3, 2018. 20

[174] G. Marcus, “The next decade in ai: four steps towards robust artificial intelligence," arXiv preprint arXiv:2002.06177, 2020. 20

[175] R. Fjelland, “Why general artificial intelligence will not be realized," Humanities and Social Sciences Communications, vol. 7, no. 1, pp. 1–9, 2020. 20

[176] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, “Performance-optimized hierarchical models predict neural responses in higher visual cortex," Proceedings of the national academy of sciences, vol. 111, no. 23, pp. 8619–8624, 2014. 20

[177] N. Kriegeskorte, “Deep neural networks: a new framework for modeling biological vision and brain information processing," Annual review of vision science, vol. 1, pp. 417–446, 2015. 20

[178] D. L. Yamins and J. J. DiCarlo, “Using goal-driven deep learning models to understand sensory cortex," Nature neuroscience, vol. 19, no. 3, pp. 356–365, 2016. 20