BONGARD-LOGO: A New Benchmark for Human-Level Concept Learning and Reasoning

Weili Nie
Rice University
wn8@rice.edu

Zhiding Yu
NVIDIA
zhidingy@nvidia.com

Lei Mao
NVIDIA
lmao@nvidia.com

Ankit B. Patel
Rice University
Baylor College of Medicine
abp4@rice.edu

Yuke Zhu
UT Austin
NVIDIA
yukez@cs.utexas.edu

Animashree Anandkumar
Caltech
NVIDIA
anima@caltech.edu

Abstract

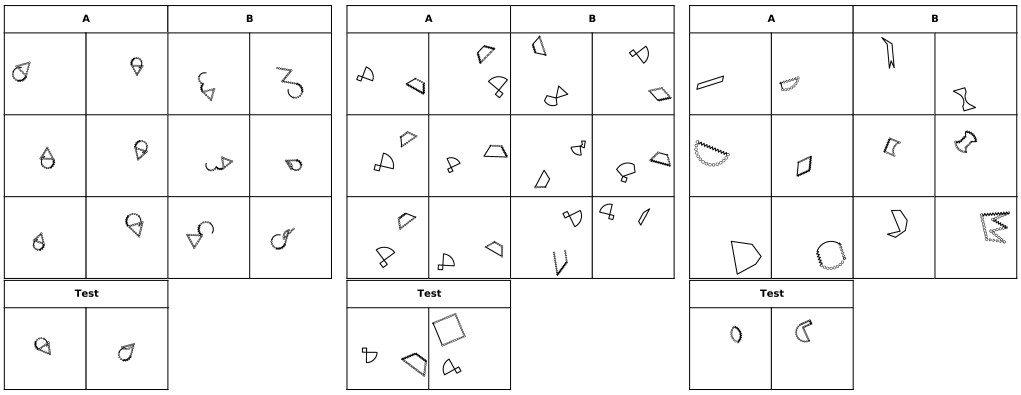
Humans have an inherent ability to learn novel concepts from only a few samples and generalize these concepts to different situations. Even though today's machine learning models excel with a plethora of training data on standard recognition tasks, a considerable gap exists between machine-level pattern recognition and human-level concept learning. To narrow this gap, the Bongard Problems (BPs) were introduced as an inspirational challenge for visual cognition in intelligent systems. Despite new advances in representation learning and learning to learn, BPs remain a daunting challenge for modern AI. Inspired by the original one hundred BPs, we propose a new benchmark BONGARD-LOGO for human-level concept learning and reasoning. We develop a program-guided generation technique to produce a large set of human-interpretable visual cognition problems in action-oriented LOGO language. Our benchmark captures three core properties of human cognition: 1) context-dependent perception, in which the same object may have disparate interpretations given different contexts; 2) analogy-making perception, in which some meaningful concepts are traded off for other meaningful concepts; and 3) perception with a few samples but infinite vocabulary. In experiments, we show that the state-of-the-art deep learning methods perform substantially worse than human subjects, implying that they fail to capture core human cognition properties. Finally, we discuss research directions towards a general architecture for visual reasoning to tackle this benchmark.

1 Introduction

Human visual cognition, a key feature of human intelligence, reflects the ability to learn new concepts from a few examples and use the acquired concepts in diverse ways. In recent years, deep learning approaches have achieved tremendous success on standard visual recognition benchmarks [1, 2]. In contrast to human concept learning, data-driven approaches to machine perception have to be trained on massive datasets and their abilities to reuse acquired concepts in new situations are bounded by the training data [3]. For this reason, researchers in cognitive science and artificial intelligence (AI) have attempted to bridge the chasm between human-level visual cognition and machine-based pattern recognition. The hope is to develop the next generation of computing paradigms that capture innate abilities of humans in visual concept learning and reasoning, such as few-shot learning [4, 5], compositional reasoning [6, 7], and symbolic abstraction [8, 9].

The deficiency of standard pattern recognition techniques has been pinpointed by interdisciplinary scientists in the past several decades [3, 10]. Over fifty years ago, M. M. Bongard, a Russian computer

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.



(a) free-form shape problem (b) basic shape problem (c) abstract shape problem

Figure 1: Three types of problems in BONGARD-LOGO, where (a) the free-form shape concept is a sequence of six action strokes forming an "ice cream cone"-like shape, (b) the basic shape concept is a combination of "fan"-like shape and "trapezoid", (c) the abstract shape concept is "convex". In each problem, set $\mathcal{A}$ contains six images that satisfy the concept, and set $\mathcal{B}$ contains six images that violate the concept. We also show two test images (left: positive, right: negative) to form a binary classification task. All the underlying concepts in our benchmark are solely based on shape strokes, categories and attributes, such as convexity, triangle, symmetry, etc. We do not distinguish concepts by the shape size, orientation and position, or relative distance of two shapes.

scientist, invented a collection of one hundred human-designed visual recognition tasks, now named the Bongard Problems (BPs) [11], to demonstrate the gap between high-level human cognition and computerized pattern recognition. Several attempts have been made in tackling the BPs with classic AI tools [12, 13], but to date we have yet to see a method capable of solving a substantial portion of the problem set. BPs demand a high-level of concept learning and reasoning that is goal-oriented, context-dependent, and analogical [14], challenging the objectivism (e.g., image classification by hyperplanes) embraced by traditional pattern recognition approaches [14, 15]. Therefore, they have been long known in the AI research community as an inspirational challenge for visual cognition. However, the original BPs are not amenable to today's data-driven visual recognition techniques, as this problem set is too small to train the state-of-the-art machine learning methods.

In this work, we introduce BONGARD-LOGO, a new benchmark for human-level visual concept learning and reasoning, directly inspired by the design principles behind the BPs. This new benchmark consists of 12,000 problem instances. The large scale of the benchmark makes it digestible by advanced machine learning methods in modern AI. To enable the scalable generation of Bongard-style problems, we develop a program-guided shape generation technique to produce human-interpretable visual pattern recognition problems in action-oriented LOGO language [16], where each visual pattern is generated by executing a sequence of program instructions. These problems are designed to capture the key properties of human cognition exhibited in original BPs, including 1) context-dependent perception, in which the same object may have fundamentally different interpretations given different contexts; 2) analogy-making perception, in which some meaningful concepts are traded off for other meaningful concepts; and 3) perception with a few samples but infinite vocabulary. These three properties together distinguish our benchmark from previous concept learning datasets that centered around standard recognition tasks [3, 7, 17, 18].

In our experiments, we formulate this task as a few-shot concept learning problem [4, 5] and tackle it with the state-of-the-art meta-learning [19–23] and abstract reasoning [17] algorithms. We found that all the models have significantly fallen short of human-level performances. This large performance gap between machine and human implies a failure of today's pattern recognition systems in capturing the core properties of human cognition. We perform both ablation studies and systematic analysis of the failure modes in different learning-based approaches. For example, the ability of learning to learn might be crucial for generalizing well to new concepts, as meta-learning methods largely outperform other approaches in our benchmark. Besides, each type of meta-learning methods has its preferred generalization tasks, according to their different performances on our test sets. Finally, we discuss potential research frontiers of building computational architectures for high-level visual cognition, driven by tackling the BONGARD-LOGO challenge.

2 BONGARD-LOGO Benchmark

Each puzzle in the original BP set is defined as follows: Given a set $\mathcal{A}$ of six images (positive examples) and another set $\mathcal{B}$ of six images (negative examples), the objective is to discover the rule (or concept) that the images in set $\mathcal{A}$ obey and images in set $\mathcal{B}$ violate. The solution to a BP is a logical rule stated in natural language that describes the visual concept presented in all images of set $\mathcal{A}$ but none of set $\mathcal{B}$. The original BPs consist of one hundred visual pattern recognition problems of black and white drawings. Through these carefully designed problems, M. M. Bongard aimed to demonstrate the key properties of human visual cognition capabilities and the challenges that machines have to overcome [11]. We highlight these key properties in detail in Section 2.2.

From a machine learning perspective, we can have multi-faceted interpretations of the BPs. Pioneer studies cast it as an inductive logic programming (ILP) problem [12] and a concept communication problem [13]. These two formulations typically require a significant amount of hand-engineering to define the logic rules or the language grammars, limiting their broad applicability. A more general and learning-oriented view of this problem is to cast it as a few-shot learning problem, where the goal is to efficiently learn the concept from a handful of image examples. We are most interested in this formulation, as it requires the minimum amount of manual specification and likely leads to more generic approaches. In the next, we introduce our benchmark that inherits the properties of human cognition while being compatible with modern data-driven learning tools.

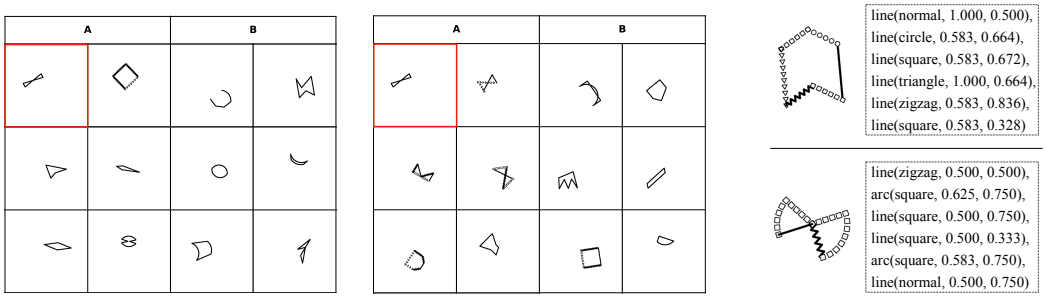
2.1 Benchmark Overview

We developed the BONGARD-LOGO benchmark that shares the same purposes as the original BPs for human-level visual concept learning and reasoning. Meanwhile, it contains a large quantity of 12,000 problems and transforms concept learning into a few-shot binary classification problem. Figure 1 shows some examples in BONGARD-LOGO. For example, in Figure 1c, set $\mathcal{A}$ contains six image patterns which are all convex shapes and, set $\mathcal{B}$ contains six image patterns which are all concave shapes. The task is to judge whether the pattern in the test image matches the concept (e.g., convex vs concave) induced by the set $\mathcal{A}$ or not. As the same concept might lead to vastly different patterns, a successful model must have the ability to identify the concept that distinguishes $\mathcal{A}$ and $\mathcal{B}$. The problems in BONGARD-LOGO belong to three types based on the concept categories:

Free-form shape problems In the first type of problems, we have 3,600 *free-form shape* concepts, where each shape is composed of randomly sampled action strokes. The rationale behind these problems is that the concept of an image pattern can be *uniquely* characterized by the action program that generates it [3, 24]. Here the latent concept corresponds to the sequence of action stokes, such as straight lines, zigzagged arcs, etc. Among all 3,600 problems, each has a *unique* concept of strokes, and the number of strokes in each shape varies from two to nine and each image may have one or two shapes. As shown in Figure 1a, all images in the positive set form a one-shape concept and share the same sequence of six strokes that none of images in the negative set possesses. Note that some negative images may have subtle differences in stokes from the positive set, such as perturbing a stroke from `straight_line` into `zigzagged_line`, and please see Appendix D.1 for more examples. To solve these problems, the model may have to implicitly induce the underlying programs from shape patterns and examine if test images match the induced programs.

Basic shape problems The other 4,000 problems in our benchmark are designed for the *basic shape* concepts, where the shapes are associated with 627 human-designed shape categories of large variation. The concept corresponds to recognizing one shape category or a composition of two shape categories presented in the shape patterns. Similar to the free-form shape problems, each problem has a unique basic shape concept. One important property of these problems is to test the analogy-making perception (see Section 2.2) where, for example, `zigzag` is an important feature in free-form shapes but a nuisance in basic shape problems. Figure 1b illustrate an instance of basic shape problems, where all six images in the positive set have the concept: a combination of "fan"-like shape and "trapezoid", while all six images in the negative set are other different combinations of two shapes. Note that positive images in Figure 1b may have zigzagged lines or lines formed by a set of circles (i.e., different stroke types), but they all share the same basic shape concept.

Abstract shape problems The remaining 4,400 problems are aimed for the *abstract shape* concepts. Each shape is sampled from the same set of human-designed 627 shape categories. But in contrast to the previous type, these concepts correspond to more abstract attributes and their combinations. The



(a) `have_four_straight_lines` (b) `have_six_straight_lines` (c) Action programs

Figure 2: (a-b) An illustration of the context-dependent perception in BONGARD-LOGO, where (a) the concept is `have_four_straight_lines`, (b) the concept is `have_six_straight_lines`. The same shape pattern (highlighted in red) has fundamentally opposite interpretations depending on the context: Whether the line segments are seen as continuous when they intersect with each other. (c) Two exemplar shapes and the action programs that generate them procedurally, where each base action (`line` or `arc`) has three arguments (from left to right): *moving type*, *moving length* and *moving angle*. Note that there are five moving types, including `normal`, `zigzag`, `triangle`, `circle`, and `square`, each of which denotes how the line or arc is drawn. Also, both values of moving length and moving angle have been normalized into $[0, 1]$, respectively.

purpose of these problems is to test the ability of abstract concept discovery and reasoning. There are 25 abstract attributes in total, including `symmetric`, `convex`, `necked`, and so on. Each abstract attribute is shared by many different shapes while each shape is associated with multiple attributes. Due to the increased challenge of identifying the abstract concept, we randomly sample 20 different problems for each abstract shape concept. Figure 1c shows an example of abstract shape problems, where the underlying concept is `convex`. Similarly, the stroke types (i.e., zigzags or lines of circles, etc.) do not impact the convexity. Large variations in all convex and concave shapes ensure that the model does not simply memorize some finite shape templates but instead is forced to understand the gist of the `convex` concept.

2.2 Properties of the BONGARD-LOGO Problems

In the BONGARD-LOGO problems, we do not distinguish concepts by the shape size, orientation and position, or relative distance of two shapes. Furthermore, the Bongard-style reasoning needs the visual concept to be inferred from its context with a few examples. All these together demand a perception that is both rotation-invariant and scale-invariant, and more importantly, a new learning paradigm different from traditional image recognition methods. In the next, we mainly focus on three core characteristics of human cognition captured by BONGARD-LOGO, and use intuitive examples to explain them: 1) context-dependent perception, 2) analogy-making perception, and 3) perception with a few samples but of infinite vocabulary.

Context-dependent perception The very same geometrical arrangement may have fundamentally different representations for each context on which it arises. For example, the underlying concept in the task of Figure 2a is `have_four_straight_lines`. Straight line segments must be seen as continuous, even when they intersect with another line segment. This does not happen, in the task of Figure 2b, where intersections have to split the straight lines to fulfill the underlying concept `have_six_straight_lines`. Another representative example is the hierarchy in concept learning. An equilateral triangle can either be interpreted as `equilateral_triangle` or as `convex`, which completely depends on what the remaining positive images and negative images are.

This property is compliant with the so-called "*one object, many views*" in human cognition [10, 25], where the existence and meaning of an object concept depend on the human understanding. Inspired by the original BPs, our benchmark contains a large number of the above examples that require context-dependent perception: $P(X, C)$, where C is the context to which an object X belongs. It does not make sense to infer the concept from X alone, as there is a need for contextual information that lies outside of X. By contrast, most current pattern recognition models are implicitly *context-free*. For example, the image classification models take a single cat image and output a cat label, without referring to any other image as context. They assume that there exist objects in reality outside of any understanding, and the goal is to find the unique correct description [15]. Therefore, the perception

in the conventional context-free models is a one-to-one mapping from any object X to its unique one representation $P(X)$. This may account for the inefficacy of current pattern recognition models, founded on the premise of context-free perception, in our benchmark.

Analogy-making perception When humans make an analogy, we interpret an object in terms of another. In such a sense, representations can be traded off – we may interpret a zigzag as a straight line, or a set of triangles as an arc, or just about anything as another thing [15, 26]. When we trade off the zigzag for a straight line, we not only interpret it as a straight line, but also project onto it all the properties of a straight line, such as straightness. Though zigzags can never be straight, one could easily imagine a zigzagged line and a zigzagged arc [26], as shown in Figure 1.

Many problems in our benchmark require an analogy-making perception. Figure 1b shows an example where the underlying concept includes `trapezoid`, even if some trapezoids have no straight lines. Instead, they are a set of circles or zigzags that form a conceptual shape of `trapezoid`. Importantly, this may not be just an issue of noise or nuisance, because when we tackle the task, we trade off some concepts for other concepts. On one hand, we have circles or zigzags as a meaningful structure for the underlying concept of free-form problems. That is, we strictly distinguish a circle shape from a triangle shape, or a zigzagged line from a straight line in free-form problems. On the other hand, we trade off a set of triangles or zigzags for a `trapezoid` (Figure 1b) in the basic shape problems, and trade off a set of circles for a `convex` concept (Figure 1c) in the abstract attribute problems. A model with analogy-making perception will precisely know when a representation is crucial for the concept and when it has to be traded off for other concept.

Perception with a few examples but infinite vocabulary Unlike standard few-shot image classification benchmarks [4, 20, 27], there is no finite set of categories to name or standard geometrical arrangements to describe in BONGARD-LOGO. Similar to original BPs, our problem set is not just about dealing with triangles, circles, squares, or other easily categorizable shape patterns. Particularly in many free-form shape problems, it would be a formidable task to explain their content to others in a context-free manner, if they have not seen the problems before. For example, Figure 1a shows a representative example of free-form problems, where we humans cannot provide the precise name of the free-form shape, as a concept, but we are still able to easily recognize the concepts by observing the strokes, and then make the right decisions on classifying novel test images [28]. As each free-form shape is an arbitrary composition of randomly sampled basic stroke structures. The space of all possible combinations makes the shape vocabulary size infinite.

The infinite vocabulary forbids a few-shot learner from memorizing geometrical arrangements in a dataset rather than developing an ability to conceptualize. This property is consistent with practical observations of human visual cognition that infer concepts from a few examples and generalize them to vastly different situations [3, 24].

2.3 Problem Generation with Action-Oriented Language

A distinctive feature of the original BPs is that a visual concept can be implicitly and concisely communicated by a comparison between two sets of image examples. It requires careful construction of the image sets $\mathcal{A}$ and $\mathcal{B}$ such that the concept can be identified with clarity. M. M. Bongard manually designed the original set of one hundred problems to convey the concepts he had in mind. However, this manual generation process is not scalable. We procedurally generated our shapes in the LOGO language [16], where a so-called "turtle" moves under a set of procedural action commands and its trajectory produces vector graphics. For each shape, the corresponding procedural action commands form its ground-truth *action program*.

The use of action programs to generate shapes has several benefits: First, with action programs, we can easily generate arbitrary shapes and precisely control the shape variation in a human-interpretable way. For example, in the generation of free-form shapes, we randomly perturb one command in the ground-truth action programs to construct a similar-looking and challenging negative set. Second, the ground-truth action programs provide useful supervision in guiding symbolic reasoning in the action space. It has a great potential of promoting future methods that use the symbolic stimuli, such as neuro-symbolic AI. Note that there exists a "*one-to-many*" relationship between the action program and shape pattern: an action program *uniquely* determines a shape pattern while the same shape pattern may have *multiple* correct action programs that can generate it. Please see Appendix C for our preliminary results of incorporating symbolic information into neural networks for better performance. More importantly, although our benchmark has an infinite vocabulary of shape patterns,

the vocabulary of base actions are of relatively small size, making the concepts compositional in the action space and much easier to generate.

To simplify the process, we use only two classes of base actions: `line` and `arc`. As each base action has three arguments: *moving type*, *moving length* and *moving angle*, an action is depicted by a function: `[action_name]([moving_type],[moving_length],[moving_angle])`, as shown in Figure 2c. Specifically for the above arguments, there are five moving types, including `normal`, `zigzag`, `triangle`, `circle`, and `square`. For instance, `normal` means the line or arc is perfectly straight or curved, `zigzag` means the line or arc is of the zigzagged-style and `triangle` means the line or arc is formed by a set of triangles. Besides, both moving length and moving angle are normalized into the range of $[0, 1]$, where 0 and 1 correspond to the minimal and maximal values, respectively. Depending on varying lengths of action programs, different combinations of these base actions form visually distinct shapes. Our benchmark contains different types of shapes, as discussed in Section 2.1, each of which requires a separate procedure in the LOGO language to generate, and we leave generation details in Appendix A.

We have open-sourced the procedural generation code and data of BONGARD-LOGO in the following GitHub repository: `https://github.com/NVlabs/Bongard-LOGO`.

3 Experiments

3.1 Methods

We consider several state-of-the-art (SOTA) approaches, and test how they behave in BONGARD-LOGO, which demands human cognition abilities. First, as each task in BONGARD-LOGO can be cast as a *two-way six-shot* few-shot classification problem, where meta-learning has been a standard framework [29, 4], we first introduce the following meta-learning methods, with each being SOTA in different (i.e., memory-based, metric-based, optimization-based) meta-learning categories: 1) *SNAIL* [19], a memory-based method; 2) *ProtoNet* [20], a metric-based method; 3) *MetaOptNet* [21] and *ANIL* [22], two optimization-based methods. As the Meta-Baseline [23] is a new competitive baseline in many few-shot classification tasks, we consider its two variants: 1) *Meta-Baseline-SC*, where we meta-train the model from scratch, and 2) *Meta-Baseline-MoCo*, where first use an unsupervised contrastive learning method – MoCo [30] to pre-train the backbone network and then apply meta-training. Besides, we consider two non-meta-learning baselines for comparison. One is called *WReN-Bongard*, a variant of WReN [17] that was originally designed to encourage reasoning in the Raven-style Progressive Matrices (RPMs) [31]. Another one is a convolutional neural network (CNN) baseline, by casting the task into a conventional binary image classification problem, which we call *CNN-Baseline*. Please see Appendix B for the detailed descriptions of different models.

3.2 Benchmarking on BONGARD-LOGO

We split the 12,000 problems in BONGARD-LOGO into the disjoint train/validation/test sets, consisting of 9300, 900, and 1800 problems respectively. In the training set and validation set, we uniformly sample problems from three problem types in Section 2.1. To dissect the performances of the models in different problem types and levels of generalization, we use the following four test set splits:

Free-form shape test set This test set (FF) includes 600 free-form shape problems. To evaluate the capability of these models in *extrapolation* towards more complex free-form shape concepts than trained concepts, the shape patterns in this test set are generated with "one longer" action programs than the longest programs used for generating the training set. Note that we consider the test set of "one longer" action programs as a basic case for extrapolation. Since this task has been shown difficult for current methods (as shown in Table 1), we did not include more challenging setups.

Basic shape test set This test set (BA) includes 480 basic shape problems. We randomly sample 480 basic shape problems in which the concept corresponds to recognizing a composition of two shapes. As each basic shape concept only appears once in our benchmark, we ensure that the basic shape test set shares no common concept with the training set. This test set is designed to examine a model's ability to generalize towards novel *composition* of basic shape concepts.

Combinatorial abstract shape test set This test set (CM) includes 400 abstract shape problems. We randomly sample 20 novel pairwise combinations of two abstract attributes, each with 20 problems. All the single attributes in this test set have been observed individually in the training set. However,

Methods	Train Acc	Test Acc (FF)	Test Acc (BA)	Test Acc (CM)	Test Acc (NV)
SNAIL [19]	59.2 ± 1.0	56.3 ± 3.5	60.2 ± 3.6	60.1 ± 3.1	61.3 ± 0.8
ProtoNet [20]	73.3 ± 0.2	64.6 ± 0.9	72.4 ± 0.8	62.4 ± 1.3	$\mathbf{65.4 \pm 1.2}$
MetaOptNet [21]	75.9 ± 0.4	60.3 ± 0.6	71.7 ± 2.5	61.7 ± 1.1	63.3 ± 1.9
ANIL [22]	69.7 ± 0.9	56.6 ± 1.0	59.0 ± 2.0	59.6 ± 1.3	61.0 ± 1.5
Meta-Baseline-SC [23]	75.4 ± 1.0	$\mathbf{66.3 \pm 0.6}$	$\mathbf{73.3 \pm 1.3}$	63.5 ± 0.3	63.9 ± 0.8
Meta-Baseline-MoCo [23]	$\mathbf{81.2 \pm 0.1}$	65.9 ± 1.4	72.2 ± 0.8	$\mathbf{63.9 \pm 0.8}$	64.7 ± 0.3
WReN-Bongard [17]	78.7 ± 0.7	50.1 ± 0.1	50.9 ± 0.5	53.8 ± 1.0	54.3 ± 0.6
CNN-Baseline	61.4 ± 0.8	51.9 ± 0.5	56.6 ± 2.9	53.6 ± 2.0	57.6 ± 0.7
Human (Expert)	-	92.1 ± 7.0	99.3 ± 1.9	90.7 ± 6.1	
Human (Amateur)	-	88.0 ± 7.6	90.0 ± 11.7	71.0 ± 9.6	

Table 1: Model performance versus human performance in BONGARD-LOGO. We report the test accuracy (%) on different dataset splits, including free-form shape test set (FF), basic shape test set (BA), combinatorial abstract shape test set (CM), and novel abstract shape test set (NV). Note that for human evaluation, we report the separate results across two groups of human subjects: *Human (Expert)* who well understand and carefully follow the instructions, and *Human (Amateur)* who quickly skim the instructions or do not follow them at all. The chance performance is 50%.

the 20 novel combinations are exclusive in the test set. The rationale behind this test set is that understanding the abstract shape concepts requires a model's *abstraction* ability, as it is challenging to conceptualize abstract meanings from large shape variations.

Novel abstract shape test set This test set (NV) includes 320 abstract shape problems. Our goal here is to evaluate the ability of a model on the *discovery* of new abstract concepts. Different from the construction of the combinatorial abstract shape test set (CM), we hold out one attribute and all its combinations with other attributes from the training set. All problems related to the held-out attribute are exclusive in this test set. Specifically, we choose "`have_eight_straight_lines`" as the held-out attribute, since it presumably requires minimal effort for the model to extrapolate given that other similar "`have_[xxx]_straight_lines`" attributes already exist in the training set.

3.3 Quantitative Results

We report the test accuracy (Acc) of different methods on each of the four test sets respectively, and compare the results to the human performance in Table 1. We put the experiment setup for training these methods to Appendix C, and the results are averaged across three different runs. To show the human performance on our benchmark, we choose 12 human subjects to test on randomly sampled 20 problems from each test set. Note that for human evaluation, we do not differentiate test set (CM) and test set (NV) and thus report only one score on them, as humans essentially perform the same kind of new abstract discovery on both of the two test sets. To familiarize the human subjects with BONGARD-LOGO problems, we describe each problem type and provide detailed instructions on how to solve these problems. It normally takes 30-60 minutes for human subjects to fully digest the instructions. Depending on the total time that a human subject spends on digesting instructions and completing all tasks, we have split 12 human subjects into two evenly distributed groups: *Human (Expert)* who well understand and carefully follow the instructions, and *Human (Amateur)* who quickly skim the instructions or do not follow them at all.

Performance analysis In Table 1, we can see that there exists a significant gap between the Human (Expert) performance and the best model performance across all different test sets. Specifically, Human (Expert) can easily achieve nearly perfect performances (>99% test accuracy) on the basic shape (BA) test set, while the best performing models only achieve around 70% accuracy. On the free-form shape (FF), combinatorial abstract shape (CM) and novel abstract shape (NV) test sets where the existence of infinite vocabulary or abstract attributes makes these problems more challenging, Human (Expert) can still get high performances (>90% test accuracy) while all the models only have around or less than 65% accuracy. The considerable gap between these SOTA learning approaches and human performance implies these models fail to capture the core human cognition properties, as mentioned in Section 2.2.

Comparing Human (Expert) and Human (Amateur), we can see Human (Expert) always achieve better test accuracies with lower variances. This advantage becomes much more significant in the abstract shape problems, confirming different levels of understanding of these abstract concepts among human subjects. Comparing different types of methods, meta-learning methods that have been specifically

designed for few-shot classification have a greater potential to address the BONGARD-LOGO tasks than the non-meta-learning baselines (i.e., WReN-Bongard and CNN-Baseline). In Table 1, we can see the better test accuracies of most meta-learning models across all the test sets, compared with the non-meta-learning models. Interestingly, WReN, the model for solving RPMs [17], suffers from the severest overfitting issues, where its training accuracy is around 78% but its test accuracies are only marginally better than random guess (50%). This manifests that the ability of learning to learn is crucial for generalizing well to new concepts.

Furthermore, we perform an ablation study on BONGARD-LOGO, where we train and evaluate on the subset of 12,000 free-form shape problems in the same way as before. As the properties of *context-dependent perception* and *analogy-making perception* are not strongly presented in these free-form problems, concept learning on this subset has a closer resemblance to standard few-shot visual recognition problems [4, 32]. We thus expect a visible improvement in their performances. This is confirmed by the results in Table 2 in Appendix C, where almost all the methods achieve better training and test performances. Specifically, the best training accuracy of methods increases from 81.2% to 96.4%, and the best test accuracy (FF) of methods increases from 66.3% to 74.5%. However, there still exists a large gap between the model and human performance on free-form shape problems alone (74.5% vs. 92.1%). It implies the property of *few-shot learning with infinite vocabulary* has already been challenging for these methods.

We then compare different meta-learning methods and diagnose their respective failure modes. First, Meta-Baseline-MoCo performs best on the training set and also achieves competitive or better results on most of the test sets. It confirms the observations in prior work that good representations play an integral role in the effectiveness of meta-learning [23, 27]. Between the two Meta-Baselines, we can see that the MoCo pre-training is more effective in improving the training accuracy but the improvements become marginal on the test sets. This implies that the SOTA unsupervised pre-training methods improve the overall performance while still facing severe overfitting issues. Second, we perform another ablation study on model sizes. The results in Figure 5 in Appendix C show that the generalization performances of different models generally get worse as model size decreases.

Besides, most meta-learning models perform much better on the basic shape problems than on the abstract shape problems. This phenomenon is consistent with the human experience, as it takes a much longer time for humans to finish a latter task. How to improve the combinatorial generalization or extrapolation of abstract concepts in our benchmark remains a challenge for these models. We also observe that each type of meta-learning methods has its preferred generalization tasks, according to their different behaviors on test sets. For example, the memory-based method (i.e., SNAIL) seems to try its best to perform well on abstract shape problems by sacrificing its performance in the free-form shape and basic shape problems. The metric-based method (i.e., ProtoNet) performs similarly to Meta-Baselines and better than memory-based and optimization-based methods. Lastly, the optimization-based methods (i.e., MetaOptNet and ANIL) tend to have a larger gap between training and test accuracies. These distinguishable behaviors on our benchmark provide a way of quantifying the potential advantages of a meta-learning method in different use cases.

4 Related Work

Few-shot learning and meta-learning The goal of few-shot learning is to learn a new task (e.g., recognizing new object categories) from a small amount of training data. Pioneer works have approached it with Bayesian inference [32] and metric learning [33]. A rising trend is to formulate few-shot learning as meta-learning [4, 5]. These methods can be categorized into three families: 1) memory-based methods, e.g., a variant of MANN [29] and SNAIL [19], 2) metric-based methods, e.g., Matching Networks [34] and ProtoNet [20], and 3) optimization-based methods, e.g., MAML [5], MetaOptNet [21] and ANIL [22]. Recent work [23, 27] has achieved competitive or even better performances on few-shot image recognition benchmarks [3, 35] with a simple pre-training baseline than advanced meta-learning algorithms. It also sheds light on a rethinking of few-shot image classification benchmarks and the associated role of meta-learning algorithms.

Concept learning Concept learning methods concern about transforming sensory observations [36] and experiences [37] into abstract concepts, which serve as the building blocks of understanding and reasoning. A rich and structured representation of concepts is programs [3, 24] that model the generative process of observed data in an analysis-by-synthesis framework. Program-based methods have been applied to tackle BPs [12, 13]. However, these methods require a substantial amount

of domain knowledge and do not offer complete solutions to original BPs. For example, [13] has relied on manually specified rules that can only solve a subset of 39 carefully selected BPs, making it infeasible for our benchmark which consists of tens of thousands of problems. Recently, hybrid concept learning systems that integrate deep learning with symbolic reasoning have gained increased attention in the research community [38, 39]. These methods offer great potential in combining the representational power of neural networks with the generalization power of symbolic operations for tackling our benchmark.

Abstract reasoning benchmarks In addition to popular machine perception benchmarks that focus on categorization and detection [1, 2], there have been previous efforts in creating new benchmarks for abstract reasoning, including compositional visual question answering [7], physical reasoning [40–42], and mathematics problems [43]. Most relevant to our benchmark, the Raven-style Progressive Matrices (RPMs) [44], a well-known human IQ, have inspired researchers to design new abstract reasoning benchmarks [17, 45]. Our benchmark is complementary to RPMs: RPMs focus on relational concepts (such as progression, XOR, etc.) while BONGARD-LOGO problems focus on object concepts (such as stroke types, abstract attributes, etc.). In RPMs, the relational concepts come from a small set of five relations [17]. In our benchmark, the object concepts can vary arbitrarily with procedural generation. Recently, [46] has proposed V-RPM that extends RPMs to real images. As V-PROM is a real image version of RPMs, it differs from our benchmark in the similar ways described above. Therefore, in contrast to RPMs where automated pattern recognition models have achieved competitive performances [47, 17], our BONGARD-LOGO benchmark has posed a greater challenge to current models due to its fundamental shift towards more human-like (i.e., context-based, analogy-making, few-shot with infinite vocabulary) visual cognition.

5 Discussion and Future Work

We introduced a new visual cognition benchmark that emphasizes concept learning and reasoning. Our benchmark, named BONGARD-LOGO, is inspired by the original BPs [11] that were carefully designed in the 1960s for demonstrating the chasms between human visual cognition and computerized pattern recognition. In a similar vein as the original one hundred BPs, our benchmark aims for a new form of human-like perception that is context-dependent, analogical, and few-shot of infinite vocabulary. To fuel research towards new computational architectures that give rise to such human-like perception, we develop a program-guided problem generation technique that enables us to produce a large-scale dataset of 12K human-interpretable problems, making it digestible by data-driven learning methods to date. Our empirical evaluation of the state-of-the-art visual recognition methods has indicated a considerable gap between machine and human performance on this benchmark. It opens the door to many research questions: What types of inductive biases would benefit concept learning models? Are deep neural networks the ultimate key to human-like visual cognition? How is the Bongard-style visual reasoning connected to semantics and pragmatics in natural language?

Prior attempts on tackling the orginal BPs with symbolic reasoning [12] and program induction [13] have also been far away from solving BONGARD-LOGO. However, along with the inherent nature of compositionality and abstraction in BPs, they have supplied a great amount of insight on the path forward. Therefore, one promising direction is to develop computational approaches that integrate neural representations with symbolic operations in a hybrid system [8, 38], that acquires and reasons about abstract knowledge in a prolonged process [48], that establishes a tighter connection between visual perception and the high-level cognitive process [14, 3]. We invite the broad research community to explore these open questions together with us for future work.

Broader Impact

Our work created a new visual concept learning benchmark inspired by the Bongard Problems. Our preliminary evaluations have illustrated a considerable gap between human cognition and machine recognition, highlighting the shortcomings of existing pattern recognition methods. In Machine Learning and Computer Vision, we have witnessed the integral role of standardized benchmarks [1, 2] in promoting the development of new AI algorithms. We would encourage future work to develop new visual cognition algorithms towards human-level visual concept learning and reasoning. We envision our benchmark to serve as a driving force for research on context-dependent and analogical perception beyond standard visual recognition. We believe that endowing machine perception with the abilities to learn and reason in a human-like way is an essential step towards building robust and

reliable AI systems in the wild. It could potentially lead to more human-interpretable AI systems and address concerns about ethics and fairness arising from today's data-driven learning systems that inherit or augment the biases in training data. A potential risk of our new benchmark is that it might skew research towards highly customized methods without much applicability for more general concept learning and reasoning. We encourage researchers to develop new algorithms for our benchmark from the first principles and avoid highly customized solutions, to retain the generality and broad applicability of the resultant algorithms.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for useful comments. We also thank all the human subjects for participating in our BONGARD-LOGO human study, and the entire AIALGO team at NVIDIA for their valuable feedback. WN conducted this research during an internship at NVIDIA. WN and ABP were supported by IARPA via DoI/IBC contract D16PC00003.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, IEEE, 2009.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[3] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[4] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2016.

[5] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017.

[6] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[7] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

[8] T. R. Besold, A. d. Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kühnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, *et al.*, "Neural-symbolic learning and reasoning: A survey and interpretation," *arXiv preprint arXiv:1711.03902*, 2017.

[9] A. d. Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, and D. L. Silver, "Neural-symbolic learning and reasoning: contributions and challenges," in *2015 AAAI Spring Symposium Series*, 2015.

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

[11] M. M. Bongard, "The recognition problem," tech. rep., Foreign Technology Div Wright-Patterson AFB Ohio, 1968.

[12] K. Saito and R. Nakano, "A concept learning algorithm with adaptive search," in *Machine intelligence 14: applied machine intelligence*, pp. 347–363, 1996.

[13] S. Depeweg, C. A. Rothkopf, and F. Jäkel, "Solving bongard problems with a visual language and pragmatic reasoning," *arXiv preprint arXiv:1804.04452*, 2018.

[14] D. J. Chalmers, R. M. French, and D. R. Hofstadter, "High-level perception, representation, and analogy: A critique of artificial intelligence methodology," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 4, no. 3, pp. 185–211, 1992.

[15] A. Linhares, "A glimpse at the metaphysics of bongard problems," *Artificial Intelligence*, vol. 121, no. 1-2, pp. 251–270, 2000.

[16] H. Abelson, N. Goodman, and L. Rudolph, "Logo manual," 1974.

[17] D. Barrett, F. Hill, A. Santoro, A. Morcos, and T. Lillicrap, "Measuring abstract reasoning in neural networks," in *International Conference on Machine Learning*, pp. 511–520, 2018.

[18] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.

[19] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *ICLR*, 2018.

[20] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical Networks for Few-shot Learning," *Advances in Neural Information Processing Systems*, 2017.

[21] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

[22] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," *ICLR*, 2020.

[23] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, "A new meta-baseline for few-shot learning," *arXiv preprint arXiv:2003.04390*, 2020.

[24] M. Lázaro-Gredilla, D. Lin, J. S. Guntupalli, and D. George, "Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs," *Science Robotics*, vol. 4, no. 26, 2019.

[25] B. Indurkhya, *Metaphor and cognition: An interactionist approach*, vol. 13. Springer Science & Business Media, 2013.

[26] K. J. Holyoak, D. Gentner, and B. N. Kokinov, "Introduction: The place of analogy in cognition," *The analogical mind: Perspectives from cognitive science*, pp. 1–19, 2001.

[27] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?," *arXiv preprint arXiv:2003.11539*, 2020.

[28] J. B. Tenenbaum and F. Xu, "Word learning as bayesian inference," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 22, 2000.

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

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[31] J. C. Raven, "Standardization of progressive matrices, 1938," *British Journal of Medical Psychology*, vol. 19, no. 1, pp. 137–150, 1941.

[32] L. Fe-Fei *et al.*, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1134–1141, IEEE, 2003.

[33] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[34] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, pp. 3630–3638, 2016.

[35] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," *ICLR*, 2018.

[36] S. K. Divvala, A. Farhadi, and C. Guestrin, "Learning everything about anything: Webly-supervised visual concept learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[37] N. Hay, M. Stark, A. Schlegel, C. Wendelken, D. Park, E. Purdy, T. Silver, D. S. Phoenix, and D. George, "Behavior is everything: Towards representing concepts with sensorimotor contingencies," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[38] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *ICLR*, 2019.

[39] C. Han, J. Mao, C. Gan, J. Tenenbaum, and J. Wu, "Visual concept-metaconcept learning," in *Advances in Neural Information Processing Systems*, pp. 5002–5013, 2019.

[40] A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, and R. Girshick, "Phyre: A new benchmark for physical reasoning," in *Advances in Neural Information Processing Systems*, pp. 5083–5094, 2019.

[41] K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "The tools challenge: Rapid trial-and-error learning in physical problem solving," *arXiv preprint arXiv:1907.09620*, 2019.

[42] E. Weitnauer and H. Ritter, "Physical bongard problems," in *Ifip international conference on artificial intelligence applications and innovations*, pp. 157–163, Springer, 2012.

[43] D. Saxton, E. Grefenstette, F. Hill, and P. Kohli, "Analysing mathematical reasoning abilities of neural models," *arXiv preprint arXiv:1904.01557*, 2019.

[44] 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.

[45] 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*, pp. 5317–5327, 2019.

[46] D. Teney, P. Wang, J. Cao, L. Liu, C. Shen, and A. v. d. Hengel, "V-prom: A benchmark for visual reasoning using visual progressive matrices," in *AAAI*, 2020.

[47] M. Kunda, K. McGreggor, and A. K. Goel, "A computational model for solving problems from the raven's progressive matrices intelligence test using iconic visual representations," *Cognitive Systems Research*, vol. 22, pp. 47–66, 2013.

[48] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, *et al.*, "Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[49] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[50] C. M. Bishop, "Mixture density networks," 1994.

Appendix

A More Details on Problem Generation with LOGO

As described in Section 2.1, three types of BONGARD-LOGO problems consist of two types of shapes. Free-form shape problems only contain *free-form shapes*, while both basic shape problems and abstract shape problems are composed of samples from 627 *human-designed shapes*. Since we have discussed how to use rendered shapes (i.e., *free-form* and *human-designed*) to form different types of problems, we will next talk about the process of generating each type of shapes.

Free-form shapes A free-form shape is generated in the following steps: 1) Decide what length of action programs a shape has. 2) Randomly and independently sample each base action and its corresponding three moving arguments (i.e., *moving type*, *moving length* and *moving angle*) in sequence, resulting in a tentative action program. 3) Execute the above tentative action program in the shape renderer (i.e., turtle graphics) to generate the shape. Sometimes, the strokes of the rendered shape are heavily overlapped with each other, making the shape difficult to recognize. If it has happened, we go back to step 2) and repeat the process until either there is no heavy overlapping or the maximum number of steps has reached. Note that for the free-form shape generation, the positive and negative images would be easily indistinguishable if the moving length and angle of each action have continuous values. Therefore, we further constrain the moving length and angle to be only sampled from a set of well-separated discrete values.

Human-designed shapes Among 627 human-designed shapes, each one is paired with a ground-truth action stokes. All these shapes are stored in a dictionary with a key-value pair: (*shape name*, *action stokes*). The only difference between *action program* and *action stokes* is that the latter does not specify the *moving type*, as both basic shape and abstract shape problems treat it as one of the nuisances. Therefore, a human-designed shape is generated in the following steps: 1) Randomly sample a shape name from a predetermined subset of 627 shape names. Note that for basic shape problems, the predetermined subset is depicted by shape categories (such as triangles, squares, etc.), while for abstract shape problems, the predetermined subset is depicted by abstract attributes (such as convex, symmetric, etc.). 2) Infer its corresponding *action stokes* by looking up the dictionary and add randomly sampled *moving type* to each action, resulting in the final action program. 3) Execute the above action program in the shape renderer (i.e., turtle graphics) to generate the shape.

During the generation process of both two types of shapes, we will randomize the initial starting point and initial moving angle of each shape, and the size of unit length, such that shape position, shape orientation and shape size are all considered as nuisances. It demands a perception that is both rotation-invariant and scale-invariant.

B More Details on Methods

In this section, we provide more details on state-of-the-art (SOTA) meta-learning approaches and other strong baselines. As BONGARD-LOGO problems can be cast as a *two-way six-shot* few-shot classification problem, where meta-learning has been a standard framework [29, 4], we first discuss different SOTA meta-learning methods:

SNAIL [19] SNAIL is a memory-based meta-learning method. It proposed a class of simple and generic meta-learner architectures that use a novel combination of temporal convolutions and soft attention, with the former to aggregate information from past experience and the latter to pinpoint specific pieces of information.

ProtoNet [20] ProtoNet is a metric-based meta-learning method. It proposed a simple method called prototypical networks based on the idea that we can represent each class by the mean of its examples in a representation space learned by a neural network. In the learned metric space, classification can be performed by computing distances to prototype representations of each class.

MetaOptNet [21] MetaOptNet is an optimization-based meta-learning method. It proposed to learn the feature representation that can generalize well for a linear support vector machine (SVM) classifier.

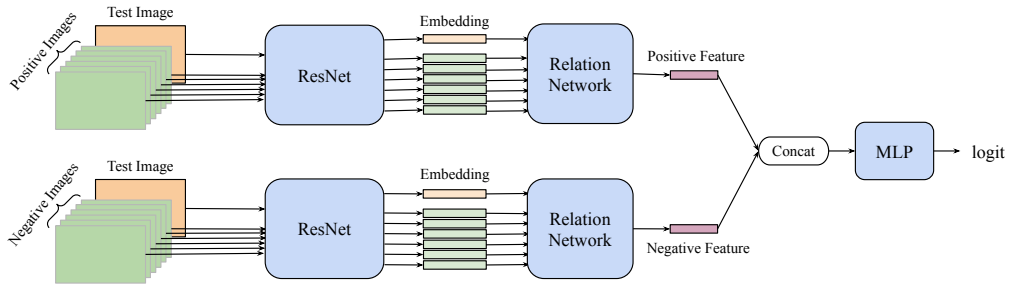


Figure 3: WReN-Bongard, where the key idea is to use relation network to form representations of pair-wise relation between each context (i.e., positive or negative) image and a given test image, and between context images themselves. Note that the 'logit' will pass into a sigmoid function for binary classification.

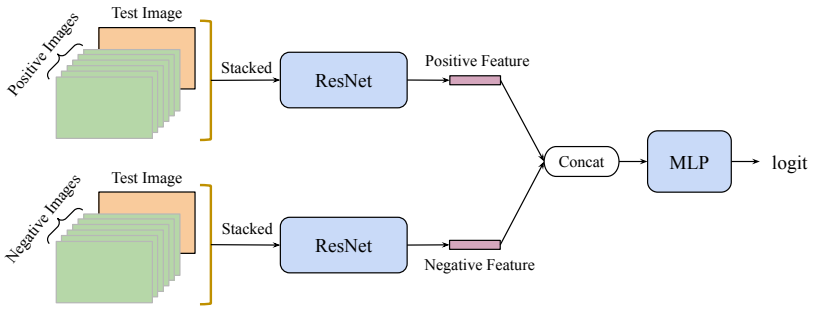


Figure 4: CNN-Baseline, where we first stack the given test image and all six positive (or all six negative images) to form a "stacked image" with input seven channels. The two stacked images pass into ResNet to extract the respective features for obtaining the final logit. Similarly, the 'logit' will pass into a sigmoid function for binary classification.

By exploiting the dual formulation and KKT conditions, MetaOptNet enabled a computational and memory efficient meta-learning with higher embedding dimensions for improved performance.

ANIL [22] ANIL is an optimization-based meta-learning method. It is a simplification of MAML [5] where we remove the inner loop for all but the (task-specific) head of the underlying neural network. ANIL matched MAML's performance on several few-shot image classification benchmarks with significantly improved computational and memory efficiency.

Meta-Baseline-SC & -MoCo Meta-Baseline [23] is a new competitive baseline designed to investigate the role of feature representations in few-shot learning. It proposed to pre-train a classifier on all base classes and then meta-learn with a nearest-centroid based few-shot classification algorithm. Because there is no base class in BONGARD-LOGO, a supervised image classification pre-training is infeasible. Instead, we introduced two variants: 1) Meta-Baseline-SC, where we meta-train the Meta-Baseline from scratch, and 2) Meta-Baseline-MoCo, where we first use an unsupervised contrastive learning method MoCo [30] to pre-train the backbone model and then apply meta-training.

Besides, we consider two strong non-meta-learning baselines for comparison.

WReN-Bongard WReN [17] is a new architecture, designed to encourage reasoning, by solving visual IQ tests, the Raven-style Progressive Matrices (RPMs) [31]. It has achieved strong performances on the RPMs. The idea of WReN is to form representations of pair-wise relation between each context and a given choice candidate, and between contexts themselves. We apply its relation network module to our problems. To do so, we develop a variant of WReN, named WReN-Bongard, by adapting WReN to Bongard problems. Figure 3 shows the architecture of WReN-Bongard.

CNN-Baseline CNN is a standard deep learning baseline of image classification without meta-learning or relation reasoning. The idea of CNN-Baseline is to stack the test image in each problem alongside all six positive images and all six negative images, respectively, to form two "stacked images" with seven input channels. This way, we convert the few-shot learning problem to a conventional binary image classification problem. Figure 4 shows the architecture of CNN-Baseline.

C More Experiment Details

Experiment setup Each image in BONGARD-LOGO is grey-scale with resolution 512×512. We train each model in Section 3.1 on the training set for 100 epochs. For a fair comparison, we use ResNet-15 (where feature map sizes are 32-64-128-256-512) with output feature dimension 128 as the backbone network in all the above methods. We use the SGD optimizer with momentum 0.9 and learning rate 0.001 with weight decay 5e-4. For all meta-learning models, we use a batch size eight on 8 GPUs, namely that each training batch contains eight Bongard problems for the loss computation. For CNN-Baseline, we use a batch size 32 on 8 GPUs. For each run of all the considered models on 8 NVIDIA V100 GPUs, it takes less than 12 hours to get the results. The other hyperparameters are kept the same with the default implementations as the respective original work.

Ablation study on BONGARD-LOGO Here we create a variant of BONGARD-LOGO, where we only include 12,000 free-form shape problems. As the properties of *context-dependent perception* and *analogy-making perception* are not presented any more, concept learning on this variant has a closer resemblance to standard few-shot visual recognition problems [4, 32]. Thus, we expect a large improvement in the performances of these methods. The results of model evaluation and human study in this variant of BONGARD-LOGO, are shown in Table 2. We can see that almost all the considered methods achieve better training and test performances. Specifically, the best training accuracy of methods increases from 81.2% to 96.4%, and the best test accuracy (FF) of methods increases from 66.3% to 74.5%. However, there still exists a large gap between the model and human performance on free-form shape problems alone. It implies the property of *few-shot learning with infinite vocabulary* has already been challenging for current methods. Another observation is that WReN-Bongard outperforms most meta-learning methods on this variant of BONGARD-LOGO, demonstrating its potential as a strong baseline for concept learning and reasoning in simpler cases.

Methods	Train Acc	Test Acc (FF)
SNAIL [19]	74.4 ± 2.5	65.2 ± 2.0
ProtoNet [20]	90.5 ± 0.6	68.5 ± 0.7
MetaOptNet [21]	91.5 ± 0.7	66.9 ± 0.5
ANIL [22]	77.8 ± 0.6	63.2 ± 0.7
Meta-Baseline-SC [23]	92.4 ± 0.3	70.8 ± 0.2
Meta-Baseline-MoCo [23]	95.8 ± 0.3	72.5 ± 0.5
WReN-Bongard [17]	$\mathbf{96.4 \pm 0.8}$	$\mathbf{74.5 \pm 4.0}$
CNN-Baseline	79.9 ± 6.2	49.8 ± 1.0
Human (Expert)	-	92.1 ± 7.0
Human (Amateur)	-	88.0 ± 7.6

Table 2: Model performance versus human performance in a variant of BONGARD-LOGO which only includes 12,000 free-form shape problems. We report the training and test accuracy (%) on the free-form shape test set (FF). Note that for human evaluation, we report the separate results across two groups of human subjects: *Human (Expert)* who well understand and carefully follow the instructions, and *Human (Amateur)* who quickly skim the instructions or do not follow them at all. The chance performance is 50%.

Capability of the CNN backbone on our visual stimuli Since the images in our benchmark are black-and-white drawings with fine lines, one may have a concern about whether the "visual recognition" part requires different capabilities from what the state-of-the-art CNN models designed for natural images. To evaluate the capability of the CNN backbone on our visual stimuli, regardless of the concept learning and reasoning aspects, we train a standard supervised recognition task with a training set of the visual inputs sampled from our benchmark. In particular, we train two separate binary attribute classifiers for `convex` and `have_two_parts`, respectively, using the same ResNet-15 backbone as in the paper. With each dataset of randomly sampled 14K positive and 14K negative samples, the model achieved the near-perfect train/test accuracies 98.7%/97.0% on `convex` and 98.0%/97.1% on `have_two_parts`. These results demonstrate that the CNN backbone is capable of processing the visual stimuli of our BONGARD-LOGO tasks.

Ablation study on model sizes To show how model sizes affect the concept learning performance, we vary the model size by dividing each layer size in the ResNet-15 backbone by a reduction factor

α, where α is a divisor of the number of parameters. Figure 5 illustrates the training and test results of two top-performing models: ProtoNet and Meta-Baseline-SC. Note that the model size decreases as α increases. We see that both training accuracies decrease consistently with smaller model sizes. On the test sets, the generalization performances also generally get worse as model size decreases, but results slightly vary across different models. ProtoNet is more sensitive to model size than Meta-Baseline-SC: (a) All the test accuracies of ProtoNet tend to decrease with smaller model sizes, while (b) test accuracies of Meta-Baseline-SC mostly remain robust to various model sizes (except for the extreme case of $\alpha = 16$).

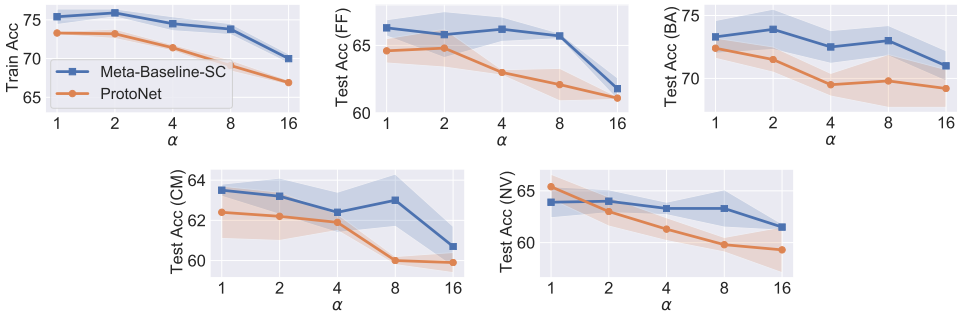


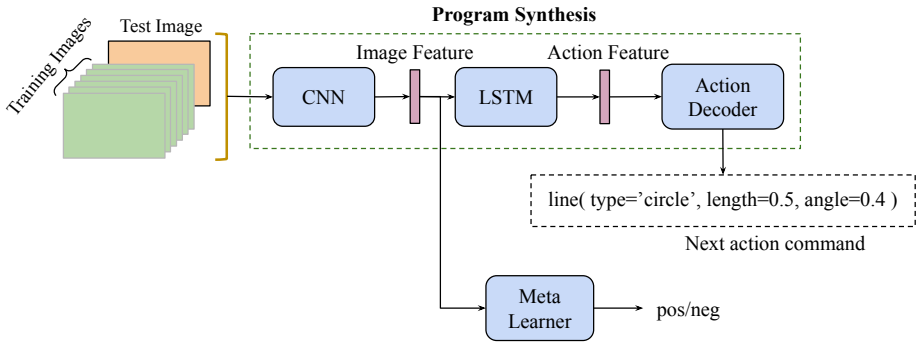
Figure 5: Performance of Meta-Baseline-SC and ProtoNet with different model sizes, controlled by a reduction factor α (i.e., the model size is smaller with a larger α).

Incorporating Symbolic Information for Better Performance Since there is a significant gap between model and human performance in our benchmark, as shown in Table 1, we have discussed the potential of neuro-symbolic approaches to close the gap in Section 5. Here we move one step forward towards a hybrid model and show some preliminary results of incorporating the symbolic information into neural networks. The basic idea is to replace the MoCo pre-training in Meta-Baseline-MoCo with the pre-training of a program synthesis task. We call the new model *Meta-Baseline-PS*, which stands for Meta-Baseline based on program synthesis.

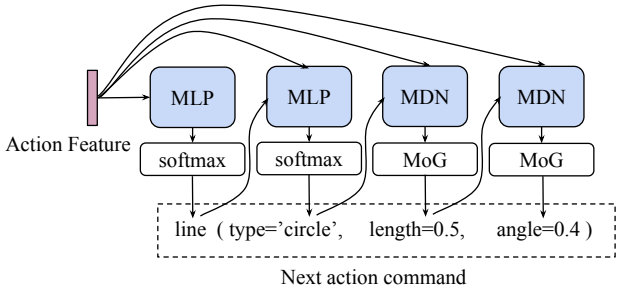
Figure 6 shows the model illustration of (a) Meta-Baseline-PS and (b) one of its components – the *action decoder*. In Meta-Baseline-PS, we first pass the input images into the CNN backbone to extract image features, which are the input of two following branches: 1) the program synthesis module where we use LSTM to convert image features into action features and then use an action decoder to synthesize the action programs; 2) the meta-learner which we use to solve the BONGARD-LOGO problems. The training of Meta-Baseline-PS is composed of two stages, i.e., we first pre-train the program synthesis module to extract the symbolic-aware image feature and then fine-tune it by training the meta-learner.

Methods	**Train Acc**	**Test Acc (FF)**	**Test Acc (BA)**	**Test Acc (CM)**	**Test Acc (NV)**
SNAIL [19]	59.2 ± 1.0	56.3 ± 3.5	60.2 ± 3.6	60.1 ± 3.1	61.3 ± 0.8
ProtoNet [20]	73.3 ± 0.2	64.6 ± 0.9	72.4 ± 0.8	62.4 ± 1.3	$\mathbf{65.4 \pm 1.2}$
MetaOptNet [21]	75.9 ± 0.4	60.3 ± 0.6	71.7 ± 2.5	61.7 ± 1.1	63.3 ± 1.9
ANIL [22]	69.7 ± 0.9	56.6 ± 1.0	59.0 ± 2.0	59.6 ± 1.3	61.0 ± 1.5
Meta-Baseline-SC [23]	75.4 ± 1.0	$\mathbf{66.3 \pm 0.6}$	$\mathbf{73.3 \pm 1.3}$	63.5 ± 0.3	63.9 ± 0.8
Meta-Baseline-MoCo [23]	$\mathbf{81.2 \pm 0.1}$	65.9 ± 1.4	72.2 ± 0.8	$\mathbf{63.9 \pm 0.8}$	64.7 ± 0.3
WReN-Bongard [17]	78.7 ± 0.7	50.1 ± 0.1	50.9 ± 0.5	53.8 ± 1.0	54.3 ± 0.6
CNN-Baseline	61.4 ± 0.8	51.9 ± 0.5	56.6 ± 2.9	53.6 ± 2.0	57.6 ± 0.7
Meta-Baseline-PS	$\mathbf{85.2 \pm 1.0}$	$\mathbf{68.2 \pm 0.3}$	$\mathbf{75.7 \pm 1.5}$	$\mathbf{67.4 \pm 0.3}$	$\mathbf{71.5 \pm 0.5}$
Human (Expert)	-	92.1 ± 7.0	99.3 ± 1.9	90.7 ± 6.1	
Human (Amateur)	-	88.0 ± 7.6	90.0 ± 11.7	71.0 ± 9.6	

Table 3: Model performance versus human performance in BONGARD-LOGO. We report the test accuracy (%) on different dataset splits, including free-form shape test set (FF), basic shape test set (BA), combinatorial abstract shape test set (CM), and novel abstract shape test set (NV). Note that for human evaluation, we report the separate results across two groups of human subjects: *Human (Expert)* who well understand and carefully follow the instructions, and *Human (Amateur)* who quickly skim the instructions or do not follow them at all. Note that Meta-Baseline-PS means the Meta-Baseline based on program synthesis, which incorporates symbolic information to solve the task. The chance performance is 50%.



(a) Meta-Baseline-PS



(b) Action decoder

Figure 6: (a) Meta-Baseline-PS, where we first pass the input images into the CNN backbone to extract image features, which are the input of two following branches: 1) the program synthesis module where we use LSTM to convert image features into action features and then use an action decoder to synthesize the action programs; 2) the meta-learner which we use to solve the BONGARD-LOGO problems. (b) the *action decoder*, the architecture for inferring action command from the action feature, where 'MLP', 'MDN' and 'MoG' stand for Multi-Layer Perceptron, Mixture Density Network and Mixture of Gaussians, respectively.

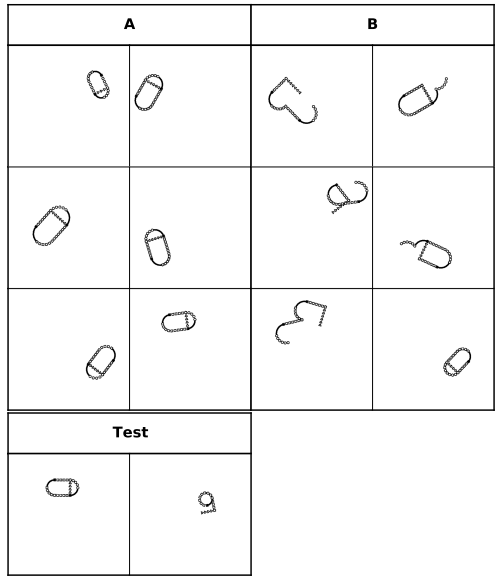
In the action decoder, the action feature from LSTM is passed into each module to sequentially predict each token in the action command, as introduced in Section 2.3. Because the values of *action name* and *moving type* are discrete, we simply use MLP and softmax to predict their values. In contrary, the values of *moving length* and *moving angle* are continuous. To learn prediction with several distinct future possibilities [49], we apply the Mixture Density Network (MDN) [50] together with the Mixture of Gaussians (MoG) parameterizations to predict their values. Experiments have shown MDN achieves better performance than the vanilla L2-norm informed prediction.

Table 3 shows the training and test of Meta-Baseline-PS on BONGARD-LOGO, along with the results of SOTA baseline approaches and human subjects. We can see that Meta-Baseline-PS largely outperforms all the SOTA baselines. It demonstrates that incorporating symbolic information into neural networks improves the overall performance, confirming the great potential of neuro-symbolic methods on tackling the BONGARD-LOGO benchmark. However, by realizing that there is still a large gap between Meta-Baseline-PS and human performance, we leave the exploration of more advanced neuro-symbolic approaches to tackle the challenge of our benchmark, as the future work.

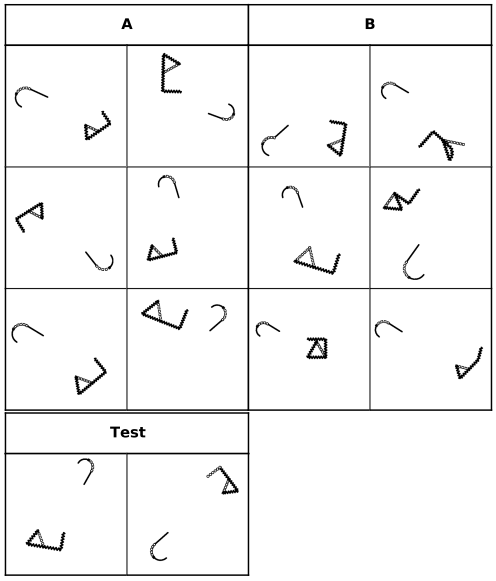
D More Examples in BONGARD-LOGO

We provide more examples from three types of problems, respectively, where each concept in any problem could be about one shape or a combination of two shapes.

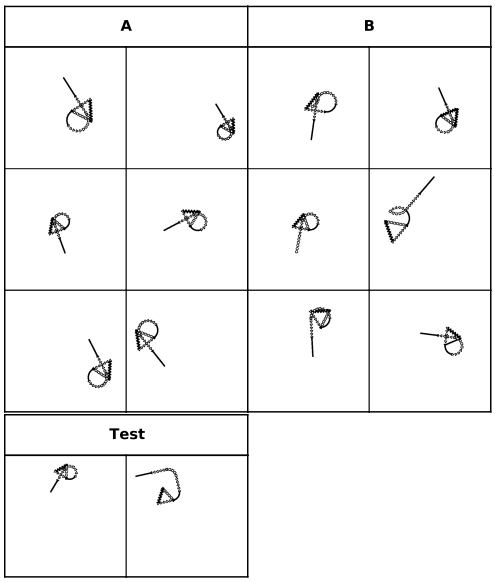
D.1 More Examples of Free-Form Shape Problems



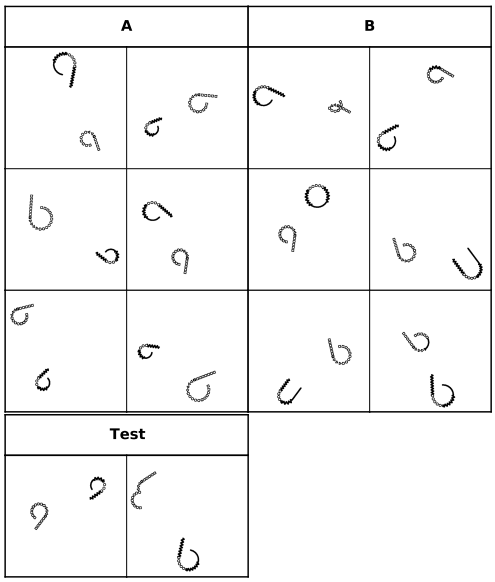
(a) # of actions: [7]



(b) # of actions: [3, 5]



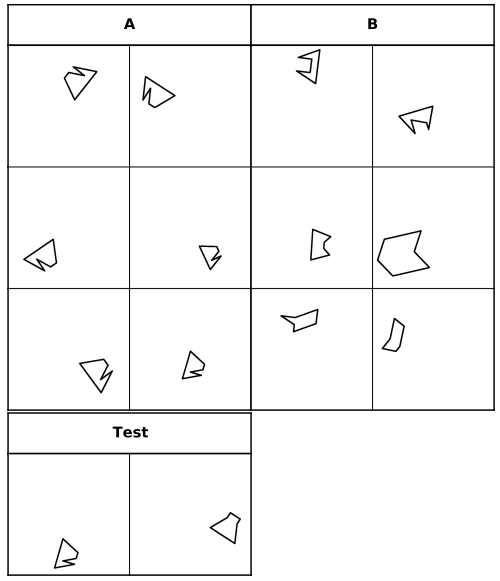
(c) # of actions: [8]



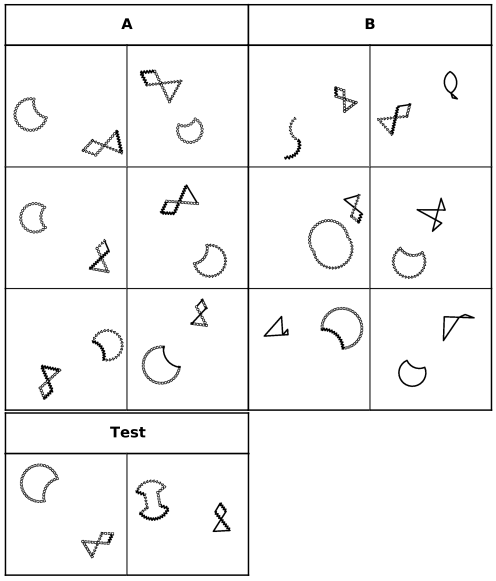
(d) # of actions: [4, 4]

Figure 7: More examples of free-form shape problems, where (a) the shape is generated by five action strokes, (b) the two shapes are generated by three and five action strokes, respectively, (c) the shape is generated by eight action strokes, (d) the two shapes are generated by four and four action strokes, respectively. In each problem, set $\mathcal{A}$ contains six images that satisfy the concept and set $\mathcal{B}$ contains six images that violate the concept. We also show two test images (left: positive, right: negative) in the binary classification problems. In free-form shape problems, the task is about discovering the concept of base strokes in a shape, which may differ in inter-stroke angles or stroke types (i.e., normal lines, zigzag lines, normal arcs, arcs formed by a set of circles, etc.). As we can see from these examples, the difference in stoke types may be subtle to distinguish. We do not distinguish concepts by the shape size and orientation, absolute position, or relative distance of two shapes.

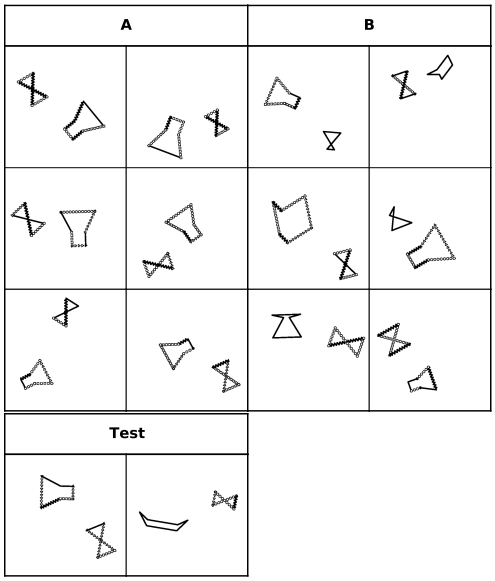
D.2 More Examples of Basic Shape Problems



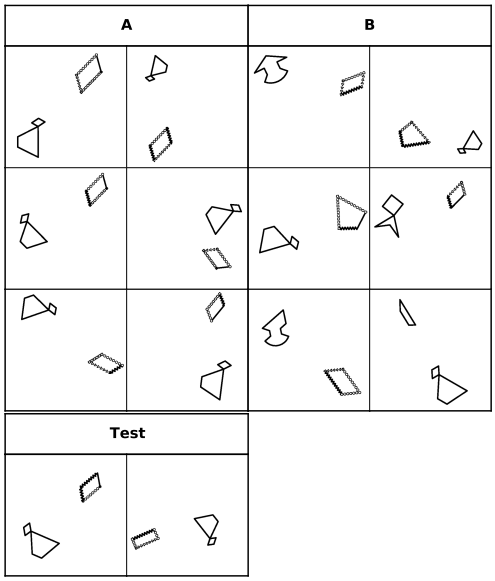
(a) `mountains`



(b) `nearly_full_moon` and `fish`



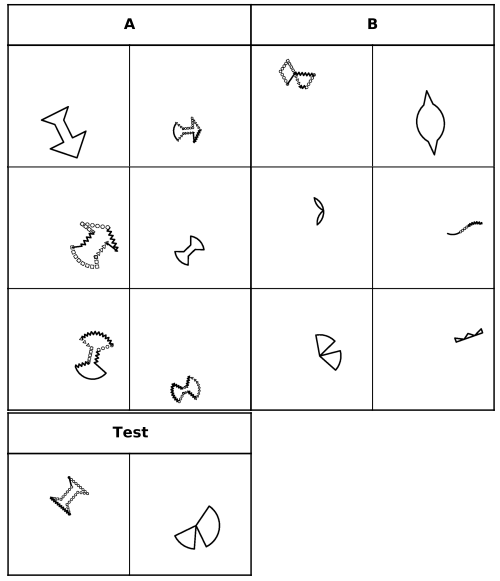
(c) `funnel` and `hourglass`



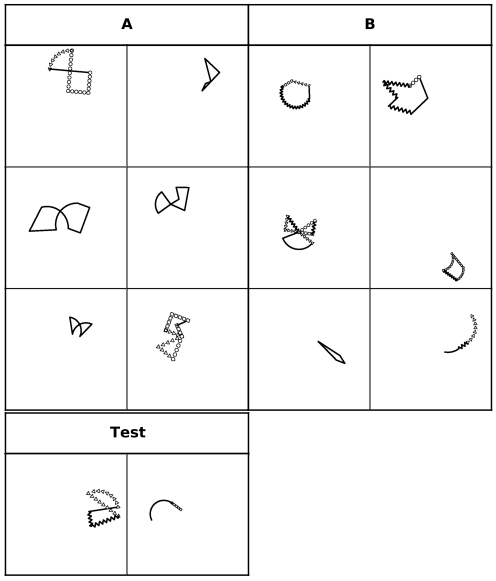
(d) `turtle` and `parallelogram`

Figure 8: More examples from basic shape problems, where (a) the concept is `mountains`, (b) the concept is a combination of `nearly_full_moon` and `fish`, (c) the concept is a combination of `funnel` and `hourglass`, (d) the concept is a combination of `turtle` and `parallelogram`. In each problem, set $\mathcal{A}$ contains six images that satisfy the concept and set $\mathcal{B}$ contains six images that violate the concept. We also show two test images (left: positive, right: negative) in the binary classification problems. In basic shape problems, the task is about discovering the concept of the shape category itself. We do not distinguish concepts by the shape size and orientation, absolute position, or relative distance of two shapes.

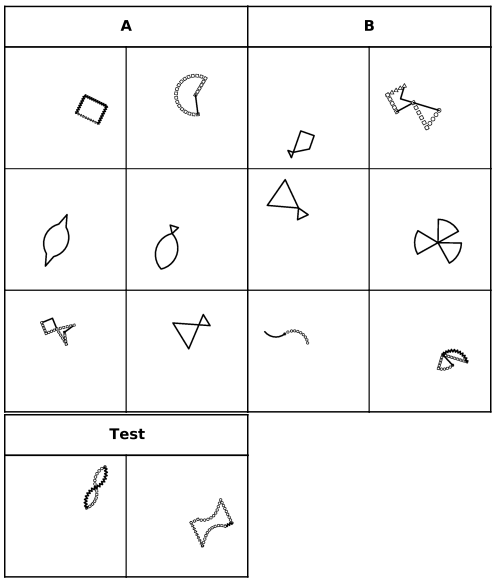
D.3 More Examples of Abstract Shape Problems



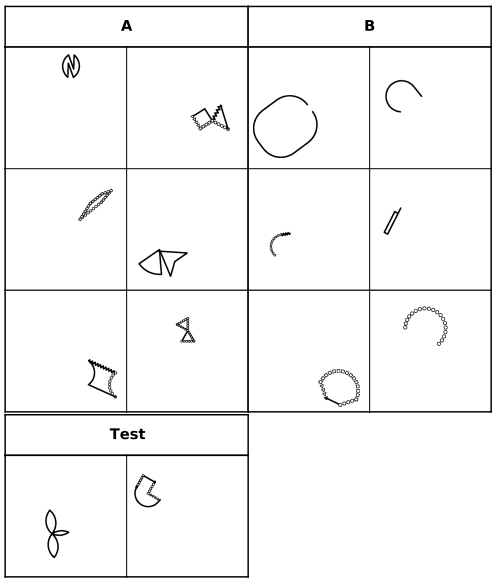
(a) `necked`



(b) `have_two_parts`

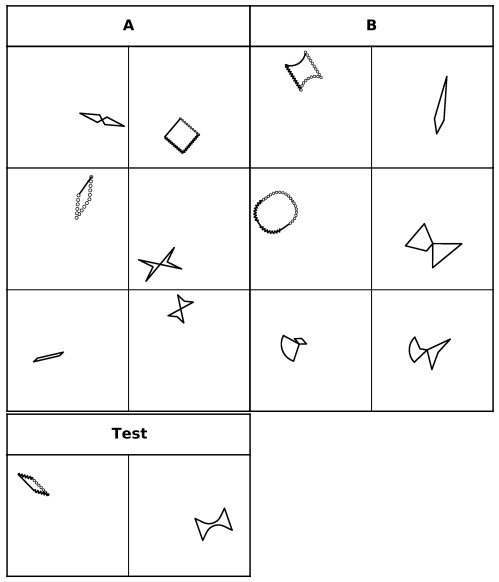


(c) `symmetric`

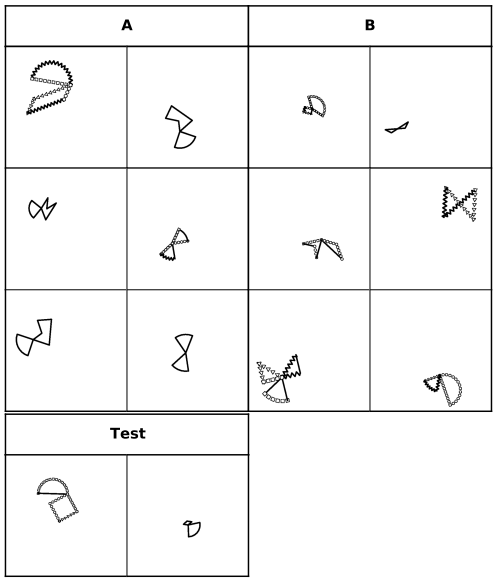


(d) `have_acute_angle`

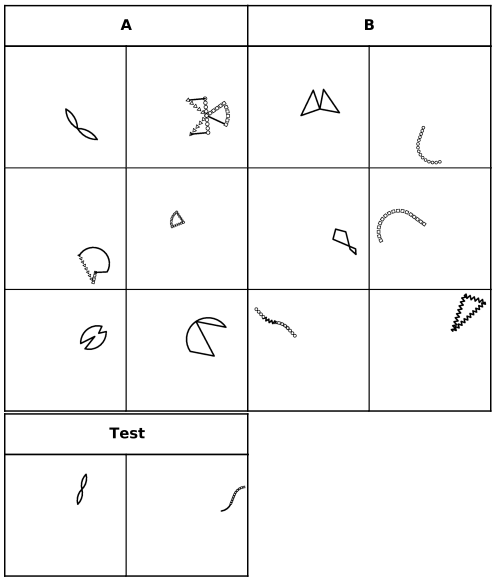
Figure 9: More examples of abstract shape problems, where one attribute shared by the positive images in set $\mathcal{A}$ is considered as the underlying concept. In particular, (a) the concept is `necked`, (b) the concept is `have_two_parts` (it means the shape can be separated into two disconnected parts by a connecting point), (c) the concept is `symmetric`, (d) the concept is `have_acute_angle`. In each problem, set $\mathcal{A}$ contains six images that satisfy the concept and set $\mathcal{B}$ contains six images that violate the concept. We also show two test images (left: positive, right: negative) in the binary classification problems. We do not distinguish concepts by the shape size and orientation, absolute position, or relative distance of two shapes. We can see for abstract shape problems, it may also be challenging for humans without clearly understanding the meanings of abstract attributes.



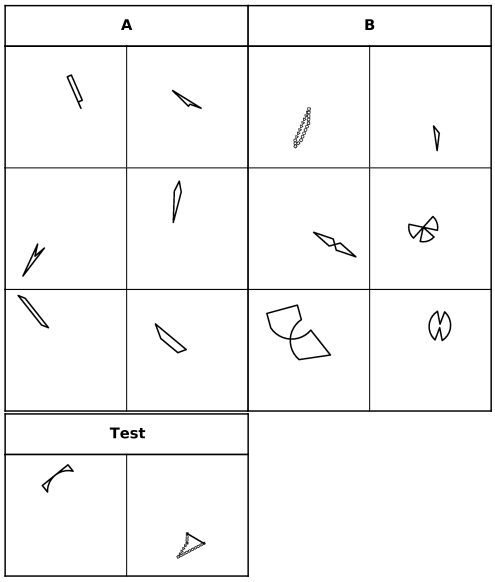
(a) `self_transposed` and `exist_quadrangle`



(b) `balanced_two` and `exist_sector`



(c) `have_curve` and `closed_shape`



(d) `have_four_straight_lines` and `thin_shape`

Figure 10: More examples of abstract shape problems, where a combination of two attributes shared by the positive images in set $\mathcal{A}$ is considered as the underlying concept. In particular, (a) the concept is a combination of `self_transposed` and `exist_quadrangle` ('self-transposed' means there is a central point, symmetrically around which every edge point of the shape could be mapped to another edge point), (b) the concept is a combination of `balanced_two` and `exist_sector` ('balanced' means the two parts in a shape have the similar area), (c) the concept is a combination of `have_curve` and `closed_shape`, (d) the concept is a combination of `have_four_straight_lines` and `thin_shape`. In each problem, set $\mathcal{A}$ contains six images that satisfy the concept and set $\mathcal{B}$ contains six images that violate the concept. We also show two test images (left: positive, right: negative) in the binary classification problems. We do not distinguish concepts by the shape size and orientation, absolute position, or relative distance of two shapes. We can see for abstract shape problems, it is also challenging for humans without clearly understanding the meanings of abstract attributes.