Psychological Review

Perception and Simulation During Concept Learning

Erik Weitnauer, Robert L. Goldstone, and Helge Ritter

Online First Publication, July 13, 2023. https://dx.doi.org/10.1037/rev0000433

AMERICAN PSYCHOLOGICAL ASSOCIATION

Psychological Review

ISSN: 0033-295X

https://doi.org/10.1037/rev0000433

Perception and Simulation During Concept Learning

Erik Weitnauer[1, 2], Robert L. Goldstone[1], and Helge Ritter[2]

[1] Department of Psychological and Brain Sciences, Program in Cognitive Science, Indiana University

[2] Center for Cognitive Interaction Technology, Bielefeld University

A key component of humans' striking creativity in solving problems is our ability to construct novel descriptions to help us characterize novel concepts. Bongard problems (BPs), which challenge the problem solver to come up with a rule for distinguishing visual scenes that fall into two categories, provide an elegant test of this ability. BPs are challenging for both human and machine category learners because only a handful of example scenes are presented for each category, and they often require the open-ended creation of new descriptions. A new type of BP called physical Bongard problems (PBPs) is introduced, which requires solvers to perceive and predict the physical spatial dynamics implicit in the depicted scenes. The perceiving and testing hypotheses on structures (PATHS) computational model, which can solve many PBPs, is presented and compared to human performance on the same problems. PATHS and humans are similarly affected by the ordering of scenes within a PBP. Spatially or temporally juxtaposing similar (relative to dissimilar) scenes promotes category learning when the scenes belong to different categories but hinders learning when the similar scenes belong to the same category. The core theoretical commitments of PATHS, which we believe to also exemplify open-ended human category learning, are (a) the continual perception of new scene descriptions over the course of category learning; (b) the context-dependent nature of that perceptual process, in which the perceived scenes establish the context for the perception of subsequent scenes; (c) hypothesis construction by combining descriptions into explicit rules; and (d) bidirectional interactions between perceiving new aspects of scenes and constructing hypotheses for the rule that distinguishes categories.

Keywords: concept learning, induction, problem solving, creativity, computational model

Humans use concepts to organize our world, make inferences, predict future outcomes, and compose complex ideas. They act as the building blocks for thought and are involved when we generalize from our previous experiences to new situations (Goldstone et al., 2018; Goodman et al., 2008; Tenenbaum et al., 2011). Although there is no shortage of models for how people acquire and use concepts (Anderson, 1991; Kruschke, 1992; Love et al., 2004; Nosofsky, 1986), there is a conspicuous gap between their restricted capacity to generalize beyond the trained examples and the highly flexible nature of people's ability to construct concepts in an open-ended manner from very few examples (Lake et al., 2015, 2017; Murphy & Medin, 1985). Furthermore, most existing models bypass the problem of grounding concepts in perceptual processes that take as input rich descriptions of objects that can be interpreted in multiple ways. They instead represent objects as fixed points in a multidimensional physical (Aha & Goldstone, 1992; Medin & Schaffer, 1978) or psychological (Nosofsky, 1984; Palmeri, 1997) space without modeling the perceptual and conceptual processes through which the stimuli are assigned their coordinates.

The goal of our perceiving and testing hypotheses on structured (PATHS) data model is to flexibly form new concepts from examples, by combining perceptual processes over rich inputs with a mechanism for building new, structured descriptions out of previously built descriptions. To achieve this, we follow four core theoretical commitments, which we believe also exemplify open-ended human category learning: (a) the continual perception of new scene descriptions over the course of category learning, (b) the context-dependent nature of that perceptual process, in which the perception of scenes establishes the context for the perception of subsequent scenes, (c) hypothesis construction by combining descriptions into explicit rules, and (d) bidirectional interactions between perceiving new aspects of scenes and constructing hypotheses for the rule that distinguishes categories.

We explore the range of these principles in PATHS for a class of learning tasks that require to infer concepts from a small set of 2D drawings of physical scenes that are challenging for humans and that have been studied in a nonphysical variant introduced under the name of "Bongard problems" (BPs; Bongard, 1970).

Robert L. Goldstone https://orcid.org/0000-0001-8357-8358

The authors would like to thank Paulo Carvalho, Peter Todd, Rich Shiffrin, Marina Dubova, Sam Gershman, and Douglas Hofstadter for their helpful discussions, and Abe Leite for extensive editing and improvements to this article and conceptualization. The code repository for the computational model described in this article, as well as the data from simulation runs, are available at https://github.com/eweitnauer/Dissertation-PATHS-Model. The behavioral results were presented in Weitnauer et al. (2014) and Weitnauer et al. (2013), and the computational model was presented in Weitnauer (2016).

Correspondence concerning this article should be addressed to Robert L. Goldstone, Department of Psychological and Brain Sciences, Program in Cognitive Science, Indiana University, Psychology Building Room 338, 1101 East 10th Street, Bloomington, IN 47405, United States. Email: rgoldsto@indiana.edu

Existing Models of Concept Induction

Given the central role that concepts play in cognition, it is hardly surprising that developing computational models of concept learning has been a major and productive research enterprise in both psychology and computer science. Much of the activity in both fields has centered on building models of *concept induction*. The inputs for models of concept induction are examples tagged with their correct concept labels. The output is a rule, characterization, or method for reliable identification of the presented inputs. Versions of this task are pervasive for children, experts, and machine learning systems. Children develop concepts of *dog*, *sticker*, and *eight* by experiencing multiple examples of each concept with a parent or teacher providing the concept label in word form (Roy et al., 2015). Much of expertise involves creating advanced concepts for the objects in one's chosen domain of expertise (Gauthier et al., 2010), such as *malignant tumor* for radiologists, *mixolydian mode* for musicians, *field* for physicists, or *car models* for automobile enthusiasts (Ross et al., 2018). Classic applications of concept induction in machine learning and artificial intelligence (AI) include diagnosing soy bean diseases (Michalski & Chilausky, 1980), classifying soil types (McBratney et al., 2003), and recommending medical treatments (Esfandiari et al., 2014).

Learning concepts from examples has been a cornerstone of AI from its beginnings in the 1960s. Inductive learning techniques come up with rule-based hypotheses based on a set of positive and negative examples, such as learning the concept of an arch from carefully crafted positive and negative examples (Winston, 1970). Inductive learning necessarily involves generalization beyond the presented examples, and T. M. Mitchell (1982) formalized generalization as a search in a typically immense space of possible hypotheses. To choose one generalization over another given that both match the training data equally well requires constraints in the learner (T. M. Mitchell, 1980). In AI, heuristics affect the order in which the space of generalizations is searched. Two high-level heuristics with several offshoots for how to search the hypothesis space are divide-and-conquer and separate-and-conquer techniques. Divide-and-conquer algorithms recursively split a data set into disjunctive sets, which are then tackled independently. Work on learning structured concepts (Hunt et al., 1966), discrimination nets (Simon & Feigenbaum, 1964), and decision trees such as ID3 (Quinlan, 1986) uses the divide-and-conquer approach. All separate-and-conquer algorithms use a similar top-level loop that searches for a rule that explains some of the positive examples, then separates these, and recursively continues the search on the remaining examples (Fürnkranz, 1999). The AI algorithms can be compared in terms of the inductive biases they impose on the language used to describe examples, the order in which hypotheses are searched, and simplicity of expressions to avoid overfitting. Another source of bias, background knowledge, is integrated with training examples into a combined deductive and inductive logical inference system as a core part of the learning process (Muggleton, 1992; Muggleton & De Raedt, 1994). Efforts have also been made to combine inductive logic programming with probabilistic and statistical inference (Dietterich et al., 2008; Getoor & Taskar, 2007).

However, even with this broadening of structured representations to include uncertainty, these systems typically ignore, or at least underemphasize, the important cognitive work needed to create new perceptual descriptions to be entered into structured descriptions of concepts. For example, when John Snow was trying to figure out the basis for categorizing 1854 Londoners as either suffering from cholera or not, he had to create a hitherto unimagined new description based on Londoners' use of water obtained from a particular, contaminated pump (Johnson, 2006). Around the same time, James Maxwell developed a conceptualization of pressure that was grounded in invisible gas molecules colliding against a container wall. Major conceptual innovations in science, mathematics, music, and art often involve constructing fundamentally new descriptions of entities in a world. Creating new concepts by coming up with fundamentally new descriptions may reach elevated peaks in scientists, artists, and mathematicians, but it is a cognitive activity engaged in by every expert and every child as well. Realizing that verbs need to be classified and used differently according to whether they refer to an imagined or real event (subjunctive: "If I *were* a rich man" vs. simple past: "When I *was* a rich man") requires learners to be able to flexibly come up with new descriptions from rich perceptual and conceptual inputs. The same is true for recognizing that frogs can be distinguished from toads based on dryness of skin. Such abilities are still conspicuously absent in AI systems (Lara-Dammer et al., 2019; M. Mitchell, 2019).

A branch of AI, constructive induction, might at first sight seem to have solved this problem, given its described objective of inventing new descriptions to be used to support concept learning (Arciszewski et al., 1995; Medin et al., 1987; Wnek & Michalski, 1994). However, a closer inspection indicates that constructive induction systems have a rather limited capacity to construct new descriptions. Given a symbolic vocabulary including *square* and *red*, these systems can construct new descriptions that are Boolean logical combinations of these elements, such as *square AND red* or *Square IF AND ONLY IF red*. There have also been proposals for automatically creating range descriptions, such that if numerosities of 12, 14, 15, 18, and 20 have all been associated with a concept, a new description of the form *12–20* may be generated (Diettrich & Michalski, 1985). Another proposal, ascending concept hierarchies, allows a system to generalize beyond *dog*, *dolphin*, and *bat* to generate the description *mammal*, if it has been provided the information that all three animals are, in fact, mammalian. However, a basic problem with these systems is that the highly simplified and idealized symbolic representations allow for very limited opportunities for flexible redescription.

A different approach for learning new descriptions comes from the burgeoning field of deep learning within machine learning. Deep learning systems are neural networks with many layers of intermediary units that transform inputs into outputs, and learning involves changing the connection strengths between layers (LeCun et al., 2015). A subfield within deep learning, representation learning, has the explicit goal of automatically creating representations from rich inputs that promote classification and prediction (Bengio et al., 2013). The representations typically take the form of internal units connecting, sometimes in long chains, inputs (e.g., 2D pixel-based encodings of many photographs) to outputs (e.g., category labels such as "German Shepard" and "airplane") that respond selectively to specific dimensions, features, or parts of the inputs. Deep learning systems have successfully acquired internal representations sufficiently sophisticated to play a better game of Go than any human (Silver et al., 2017) and classify skin lesions into medical categories at human levels of accuracy (Esteva et al., 2017). Impressively, some of these

systems can disentangle inputs into component dimensions that are readily interpretable by humans. For example, hand-drawn digits can be decomposed into dimensions corresponding to the identity of the digit, its rotation, and width (Chen et al., 2016). Likewise, faces can be decomposed into their identity, displayed emotion, and orientation (Guo et al., 2016). Such representations often arise when encoding the images in a low-dimensional space such that images can be faithfully reconstructed from their encodings (Higgins et al., 2017). This approach is quite interpretable due to the constraint that the low-dimensional encodings are to follow a standard normal distribution: It is easy to see in which way a particular input might be exceptional.

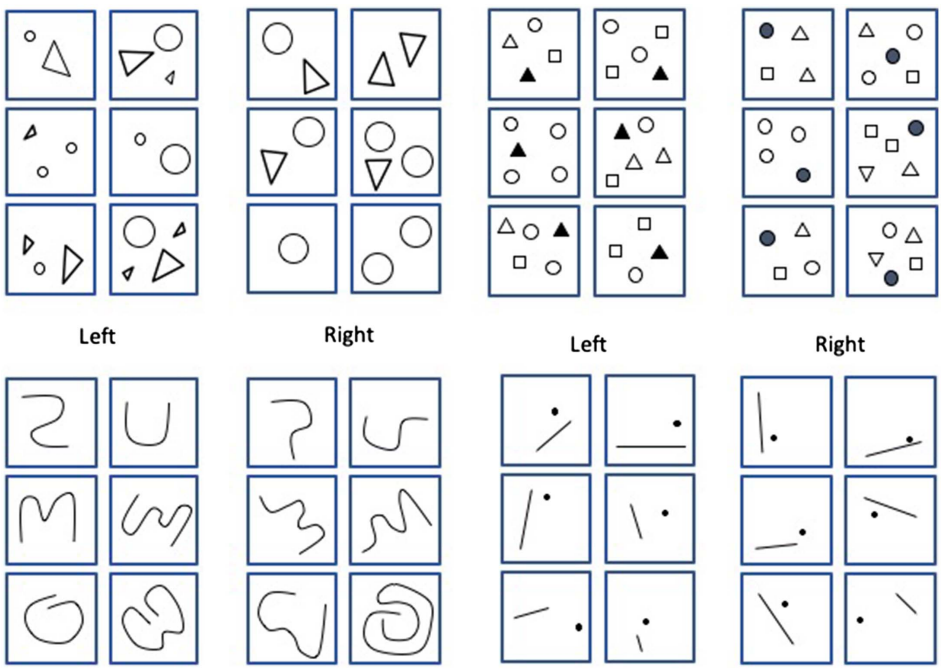
Two shortcomings with initial efforts to learn new representations with deep learning systems are the large amount of training needed to acquire effective internal representations, and the lack of explicit, structural descriptions that can be entered into generative, compositional expressions (Lake et al., 2017). Recent efforts have tackled both of these challenges. One- and few-shot learning algorithms are able to learn classifications from very small training sets by augmenting standard back-propagation deep learning with memory for stored instances (Webb et al., 2021) and attentional highlighting of particularly diagnostic features or spatially compact parts (Zhang et al., 2018). Another augmentation of standard deep learning systems has been proposed to enable these networks to learn relational descriptions. Such descriptions are required to answer questions like "Are there any blue things that are the same size as the yellow tall cylinder?" when presented with an accompanying image of a complex scene containing colored geometric objects (Vinyals et al., 2016). Relational learning networks can learn to classify images of everyday objects or novel characters from just a few examples (Sung et al., 2018).

BPs

BPs offer an elegant domain for concretely exploring issues related to concept induction. Mikhail Bongard (Bongard, 1970) proposed that a challenging task for future AI systems would be to determine the rule that distinguishes between two categories, with the input being six examples belonging to each category. Each example is a scene made up of visual elements, typically abstract shapes, lines, and forms. Six scenes are positioned on the left side and are members of one category; the other six scenes are positioned on the right side and belong to a second category. Figure 1 shows four examples of BPs, with the reader invited to try to determine a rule for distinguishing the left versus right scenes (solutions are in Appendix A). Bongard provided 100 such problems of increasing difficulty. Hofstadter (1979) introduced BPs to a wide audience and created many new BPs, and the Phaeaco model was subsequently built to solve some of them (Foundalis, 2006). Impressively, Phaeaco's architecture incorporates both low-level perceptual processes working at the pixel- and high-level symbolic analogical reasoning processes. Since then, additional systems have been proposed to solve BPs, including both deep learning neural network (Yun et al., 2020) and more symbolic (Depeweg et al., 2018) approaches.

BPs have a number of useful properties for comparing human and machine problem solving. First, because they generally involve abstract visual forms, they require few culturally specific sources of

Figure 1
Four Bongard Problems



Note. The task is to find a rule that distinguishes the six panels on the left from the six panels on the right. Solutions are provided in Appendix A. See the online article for the color version of this figure.

knowledge that a general problem solver would need to have learned. While sensitivity to some geometric properties may depend upon education and immersion in a particular culture, the kinds that feature prominently in BPs are frequently noticed by individuals coming from very disparate cultures (Dehaene et al., 2006). Second, BPs crucially involve the creation of new descriptions that are unlikely to be part of the initial representation that an observer forms when seeing a scene. For example, to solve the lower right problem in Figure 1, the observer needs to create a description that explicitly relates the lengths of two lines (one actual and one that must be projected by the solver) within a scene—something like *equal* (*length(Line1), length(Line2)*). Given the large number of potential relations that could be read off of a scene, it is unlikely that all of these are part of the observer's initial description (Hummel & Holyoak, 1997). As such, these problems allow us to explore the key process of computing new descriptions in order to characterize a category. Third, unlike the massive amount of training examples required by many deep learning systems, solving BPs requires learning category descriptions from only a few examples per category. Machine learning approaches that can learn from only a few examples, "few-shot learning systems," usually succeed by having strong constraints on the kinds of hypotheses that they form. For example, in the Omniglot project (Lake et al., 2015), previously unknown handwritten characters can be learned and generalized to new instances because characters are assumed to be generated by motions constrained to be easily produced by hands. This combination of open-ended description creation and very limited number of training instances makes BPs a paradigmatic case of inductive learning. People have a remarkable ability to create new descriptions from very few examples if the examples are chosen for their diagnosticity and capacity to eliminate ambiguity with respect to satisfactory rules.

More broadly, BPs provide a fertile testing ground for theories of creativity. Most of the component processes involved in creative problem solving, such as information gathering, conceptual combination, idea generation, and idea evaluation (Mumford & McIntosh, 2017), are core to solving BPs. The open-ended nature of BPs also fits a growing interest in infant (Twomey & Westermann, 2018) and machine learning (Stanley & Lehman, 2015) that is governed by curiosity-driven exploration rather than maximization of an objective function. Solving BPs often involves simply trying to notice new things in a scene without directly engaging in hypothesis testing.

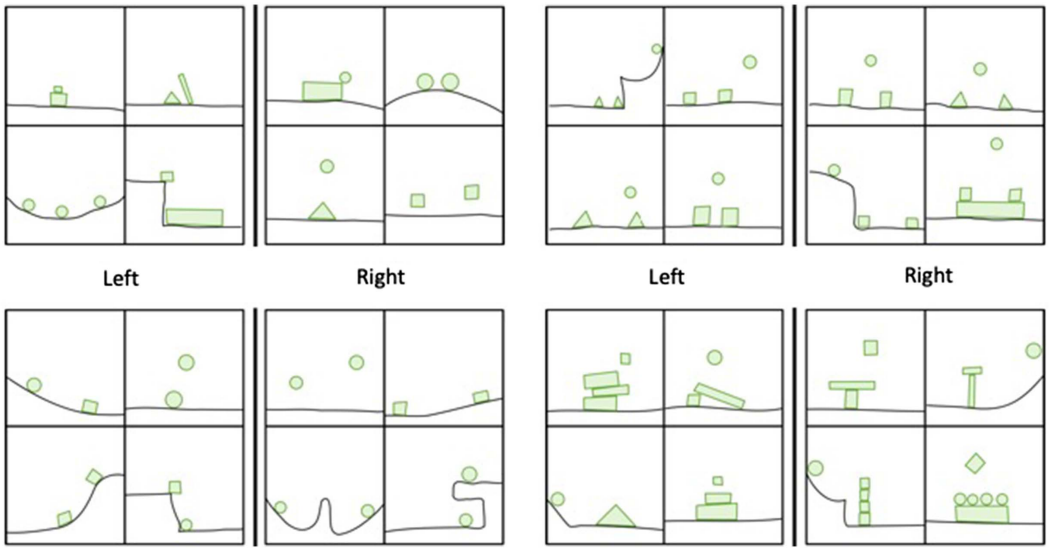
We introduce a new kind of BPs that we call physical Bongard problems (PBPs). In these problems, constraints on the content of the scenes are introduced in order to shift the focus from low-level visual processing toward dynamics, simulation, and interaction. To solve a PBP, the solver must perceive and predict the spatial dynamics in the depicted physical scenes. This predictive aspect of perception is essential for embodied agents that interact with a dynamic world in general and is strongly present in human cognition (Clark, 2013; Hubbard, 2005). To solve PBPs, assumptions are made about a mass associated with each object and the presence of a downward-directed gravitational force. Recent research indicates that people often dynamically simulate scenes using an internal model that is roughly comparable with a virtual physics engine to draw inferences (Allen et al., 2020; Battaglia et al., 2013; T. D. Ullman et al., 2018). Consistent with this work, PBPs require physical interactions to be simulated to see common properties shared by scenes belonging to the same category. For example, the scenes may contain arbitrary nonoverlapping rigid objects that could stand stably on the ground, be positioned in midair, or be placed on the side of a steep hill. The objects are understood to be at rest at the time of the initial scene and no hidden joints or self-propelled objects are allowed. We have created 36 PBPs, which explore different feature and relation types as well as different solution structures. They are based on static or dynamic object properties such as "shape" and "stability," on spatial and physical relations between objects such as "left of," "close to," and "supports," as well as properties of groups of objects or whole scenes such as numerosity and predictability. Some problems focus on events that happen at a particular time during the imagined unfolding of events like collisions between objects, whereas others are based on the reaction of objects to a simple kind of imagined interaction, like pushing or lifting an object.

Four examples of PBPs are shown in Figure 2. Our reasons for introducing PBPs as a special, hitherto unconsidered case of BPs are threefold. First, because the would-be solver needs to simulate the events within a scene to find the discriminating rule, PBPs make it particularly clear that the features that enter into these rules are not immediately available when a scene is presented to the solver. For example, for the upper right BP in Figure 1 it might be tempting to assume that features such as *black* and *circle* are immediately registered and part of the initial, default representation for the scene. In contrast, it is clear that the solution-relevant description *landing location between two identical objects* cannot be immediately read off of the scenes in the upper right BP of Figure 2. Generating this description requires an internal simulation to be run that obeys gravity and physical laws of interaction between objects.

Second, because the solutions to PBPs require internal simulations that are computationally costly, they emphasize the need to prioritize the computation of descriptions. While *circle* may be instantaneously available to a concept induction system if it is part of the encoding of a scene, the description *landing location between two identical objects* can only be ascribed to a scene after a costly computation, thereby highlighting the need to prioritize the derivation of descriptions. For the PATHS model that we describe next, every scene in a PBP establishes a context for every other scene. As a result, descriptions as they are created for one scene affect the prioritization of descriptions in other scenes.

Third, PBPs require flexible interpretive processes that are a hallmark of adaptive intelligence and are still challenging for modern machine learning systems. There have recently been striking advances in systems that can generate richly and compositionally structured images from text descriptions (Ramesh et al., 2022), learn how to improve their own learning across tasks (Flennerhag et al., 2022), and appropriately respond to a broad range of natural language queries (Bubeck et al., 2023). Despite these remarkable successes, PBPs provide a challenging testbed for cognition because (a) scenes that belong to one category look superficially similar to scenes that belong to the other category; (b) highly specific interpretations and simulations are needed to solve a categorization problem; (c) novel PBP problems can be created, even automatically, that are not in any preexisting training set; and (d) because of the difficulty in precomputing all of the possible interpretations of a scene that might be involved in a categorization rule, it is practically necessary for a successful system to flexibly generate new interpretations of a scene during problem solving. Technological

Figure 2
Four Example Physical Bongard Problems



Note. In these examples, each problem has four scenes in the left category and four scenes in the right category. The two categories are distinguished by a rule that assumes that the objects in the scenes follow a natural two-dimensional physics. Solutions are provided in Appendix A. See the online article for the color version of this figure.

breakthroughs in AI have underscored the need to develop assessments that measure adaptive, humanlike intelligence. PBPs offer a promising addition to these assessments because it is hard to imagine a system that can solve the wide variety of PBPs that humans can solve without also possessing concepts (M. Mitchell & Krakauer, 2023).

PATHS shares with previous works the goal of computational modeling of concept formation based on small sets of examples of 2D scenes. At the same time, PATHS makes a strong shift toward dynamics, simulation, and interaction. This leads to major distinctions from previous approaches that develop a fixed pipeline of feature generation steps applied to a static scene input. We will sketch these differences only briefly here (postponing a fuller discussion until Section "Comparison of PATHS with other Bongard Solvers" on p. 46). In the seminal Phaeaco work, this pipeline builds a relational scene graph from which a discriminative rule is derived. In Depeweg et al. (2018), the Phaeaco scene parser is replaced by a set of visual routines that compute a set of predetermined basic geometric features (such as position, size, or elongation). A hand-crafted grammar is used to derive from them higher level features that are used to generate a decision tree for the final discrimination.

While this framework improves on the previous Phaeaco approach in terms of solved BPs, its lack of dynamics and the absence of any processes that use physical simulation to predict scene changes prevent it from successfully handling most PBPs. The same applies to a more recent approach (Yun et al., 2020), which employs a pretrained convolutional neural network (CNN) for feature generation and substitutes the symbolic grammar by either a one-level classification tree or a regression layer. In addition, the use of features generated by deep artificial neural networks also prevents the autonomous construction of human-readable rules. Rules are only derived post hoc by human inspection of averaged network activation maps.

The PATHS Model for Solving PBPs

PBPs abstract from the complexity of real-world scenes, yet they are sufficiently complex to require attentional processes to prioritize the ongoing process of producing scene descriptions. They also require perceiving the future of a scene, including predicting the paths of moving objects and predicting how a specific configuration of objects might behave over time or react to external forces. Two necessary component processes for open-ended induction in PBPs are visual attentional processes for identifying features and relations in scenes and the construction of goals and hypotheses, which, when formed, will guide subsequent visual attentional processes.

To solve open-ended induction problems such as PBPs, PATHS is built around four core theoretical commitments. First, creating new perceptual descriptions proceeds concurrently with rule construction and testing. While it may be tempting to first perceive scene features and then use these features to form rules, there are too many candidates descriptions that are possible to make this feasible. Second, what perceptual descriptions are constructed for a scene depends upon the other descriptions created for other scenes. Third, descriptions are composed of structured representations to produce rules that distinguish between sides of a PBP. Fourth, there are bidirectional influences between perceiving descriptions and constructing rules. These four principles are further elaborated in the following section. We then describe the basic constituents that PATH uses for their implementation and conclude this section.

Underlying Principles

This section describes how the perceptual capabilities of the PATHS model are integrated with a hypothesis generation and testing mechanism. The PATHS model perceives features on objects, object pairs, and groups of objects in PBPs and uses these perceptions to construct structured, rule-based representations of the scenes. The design of the PATHS model was guided by a number of principles meant to capture aspects of human cognition while solving PBPs. These principles are the following.

The Continual Perception of New Scene Descriptions

Over the course of solving a PBP, PATHS continues to generate new scene descriptions. This stands in contrast to the typical use of exhaustive initial encodings made in categorization models. The reason why exhaustive initial descriptions are insufficient for PATHS is that the perception of attributes and relations, which often involves mental simulations of physics, requires time and effort. In order to learn efficiently, PATHS uses feature and object saliencies, as well as information from hypothesis-to-scene matches, to decide what to next perceive. After perceiving a couple of scenes in a PBP, PATHS will have noticed some reoccurring patterns and will have created some solution hypotheses. These influence the choice of what to perceive next.

The Context-Dependent Nature of That Perceptual Process

The membership of an instance to a concept is often graded and context dependent. For example, when perceiving how "close" two objects are to each other, they can be very close, close, or not so close. At the time, the model formulates a rule based on feature membership values, these values are discretized into an all or nothing membership (close or not close). However, the thresholds that distinguish between these membership states can be adjusted depending on the context. The same object might be called "small" in one context and "big" in another.

Hypothesis Testing

The PATHS model learns rules that sort structured instances into categories through an active process of constructing and testing hypotheses. The model's rule space is restricted to conjunctions of object attributes, group attributes, and object relations. The rules can be all-, exists-, and unique-quantified over objects. PATHS works iteratively on the scenes and is, like humans, influenced by the order in which the examples are presented.

Bidirectional Interactions Between Perceiving New Aspects of Scenes and Constructing Hypotheses for the Rule That Distinguishes Categories

The processes of perceiving PBP scenes and constructing rule-based interpretations of them happen at the same time and influence each other. The model starts working on PBPs without knowledge of any object attributes or spatial relations. Instead, by detecting features and positions of the objects, the model perceives features step by step in order to build scene descriptions and hypotheses.

Overview of How the PATHS Model Solves PBPs

Solving PBPs in PATHS is a dynamical process arising from suitable interactions between the following key entities in PATHS:

Objects

Unlike Phaeaco (Foundalis, 2006), PATHS does not take as input a pixel-based bitmap representation of scenes. Rather, its input corresponds to a scalable vector graphics (SVGs) image in which a scene is composed of prearticulated objects with known feature values. Each of these prearticulated objects is considered a discrete entity by PATHS. Our main interest regarding the perceptual processes involved in solving PBPs is not in the detection of objects, which is largely solved by current image recognition algorithms. Instead, we focus on the perception of dynamic and relational aspects of each scene. To this end, SVGs provide a useful level of abstraction that makes the positions and outlines of objects and ground readily available to PATHS.

Groups

PATHS can consider groups of objects. These groups are united by proximity or a common feature selector.

Features

PATHS can perceive certain features of objects and relations between objects. These include (a) geometric features, (b) spatial relations, and (c) physical features and relations. Features are not necessarily binary. For example, two objects might be very close, somewhat close, or not close at all. Such "degree-like" features are expressed as fuzzy membership values in the interval [0, 1]. The types of features and their perception will be discussed at length in the section on how PATHS perceives physical scenes.

Selectors

A selector is a binary test on an object or group of objects that assesses the extent to which it exhibits a feature or some combination of features. The simplest selectors consist of a single feature and a threshold, but selectors can be combined and refined. The transition from perceptions to selectors will be discussed in the section on how PATH builds physical concepts.

Hypotheses

A hypothesis consists of a selector, a quantifier, and a side. An example hypothesis is "all objects on the right side are not moving." The aim of PATHS is to construct a hypothesis that is consistent with all of the exemplars on one side of the PBP and none on the other. The construction and application of hypotheses are discussed in the section outlining the core loop of PATHS.

Actions

At each step, PATHS takes an action in order to develop more perfect hypotheses. The voluntary actions are (a) perceive, (b) check hypothesis, and (c) combine hypothesis. Combine hypothesis has subactions: (a) combine selector and (b) refine relative selector.

There is also the involuntary action create hypothesis, which is triggered following a perception. The different actions will be discussed deeply in the section outlining the core loop of PATHS.

The source code of the computational model is openly available on Github at https://github.com/eweitnauer/Dissertation-PATHS-Model. Readers can explore an interactive version of the model in Chrome (other browsers may not work) at https://graspablemath.com/portfolio/paths/sites/public-0.7.1/index.html. This implementation is preloaded with several PBPs and also allows users to create their own PBPs by dragging SVG files into the boxes that comprise a PBP. The objects and the ground in each scene are represented as polygons that describe their outline, and, through their shading, whether they are static or dynamic. We now turn to describe how the above key entities interact in PATH to generate solutions for given PBPs and will end with a walk-through of an actual PBP solution attempt.

High-Level Description of PATHS's PBP Solving Process

When the model starts working on a PBP, the first step is to load all the scenes that are provided as SVG images into memory. Throughout the PBP solving process, PATHS only sees two of the scenes at a time, which is not necessarily from different sides of the PBP. PATHS only works on the currently visible scene pair and proceeds through a predefined sequence of scene pairs, to align with human experimental conditions. Initially, the model knows nothing about the objects other than their existence and starts gathering information about the objects in the first visible scene pair. It selects features, such as "large" or "stable," and objects on which to perceive the features. After a new perception is made, a corresponding selector, such as "large > 0.5," is created. This selector is then applied to both scenes in the currently visible scene pair, potentially resulting in a number of objects in both scenes that match. The match results and the selector are both captured in a hypothesis, which represents a potential solution or part of a solution.

After some perception steps, the model switches to the next scene pair. It can now continue to perceive features on the new objects or check existing hypotheses on the new scenes to gather additional evidence about their likelihood. A third type of action is to combine existing hypotheses to build more complex hypotheses. For example, "large objects" and "small objects on top of any object" can be combined into "small objects on top of large objects." The model stops as soon as a hypothesis has been checked on all scenes and is determined to be a solution, which means it matches all scenes from one side and no scene from the other side. This provides an implicit bias toward parsimonious hypotheses.

PATHS determines the next action by randomly drawing from a fixed multinomial distribution. The elements that the chosen action is acting on are determined stochastically based on information from all hypotheses thus far formed. More promising hypotheses will be more likely to be checked first. Objects and features that play a role in promising hypotheses will be picked with a higher probability of perceiving further features.

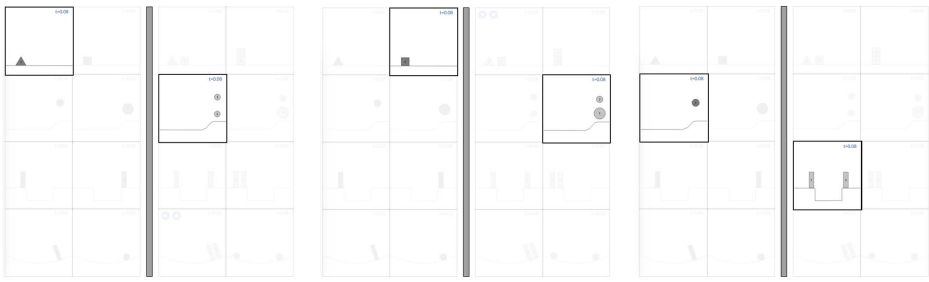
Example of a PATHS Run

We now consider a run of PATHS solving PBP02 (see Appendix B). The solution to PBP02 is that the left scenes have one object, whereas the right scenes have two objects, making it one of the most basic PBPs.

Scene Pair 1

On one particular run, PATHS started in the situation depicted in Figure 3a, with the first scene pair visible. In the sequence used for this run, each scene pair always contains scenes from different sides of the PBP. When referring to scenes, *X*–*Y* refers to the *X*th row and *Y*th column, where scenes in Columns 1 and 2 belong to the left category and Columns 3 and 4 belong to the right category. The first few actions the model took were to perceive how "large" the object in Scene 1–1 was. The model then perceived how "close" the two objects in Scenes 2–3 were, determined that they were fairly close, and subsequently created a respective pattern description ("close to any object"). This was turned into a first Hypothesis 1 that was tested on both visible scenes and stored as "only in the right scenes does there exist an object that is close to another object." Influenced by

Figure 3
The First Three Scene Pairs PATHS Looked at During a Run on PBP02



(a) 1st scene pair (b) 2nd scene pair (c) 3rd scene pair

Note. The model probabilistically decides whether to progress to the next pair of scenes through a fixed sequence of scene pairs that matches sequences given to human participants. PATHS = perceiving and testing hypotheses on structures; PBP = physical Bongard problem. See the online article for the color version of this figure.

the fact that this is a potential solution, the model then switched to the next scene pair.

Scene Pair 2

PATHS continued to work on the second scene pair as shown in Figure 3b. It decided to check Hypothesis 1 on the new scenes and found that it matched. PATHS then noticed that Object 2 (the small circle) in Scenes 2–4 "moves" and created and checked Hypothesis 2 "only in the right scenes does there exist a moving object."

Scene Pair 3

After switching to the next scene pair, as shown in Figure 3c, PATHS checked Hypothesis 1 on the new scenes and found a mismatch. Since Hypothesis 1 matched neither the left nor the right scene, it was excluded from the hypothesis list. Then, the model perceived that the right rectangle from Scene 3–3 is rectangular, which leads to a corresponding Hypothesis 3 "only in the right scenes does there exist a rectangular object." Although PATHS has already seen scenes that disconfirm this hypothesis, this hypothesis will not be eliminated until it is seen to be violated in subsequently presented scenes.

Scene Pairs 4–8

PATHS proceeded to look at the scene pairs four to eight, perceiving features, creating and checking hypotheses, and on some occasions unsuccessfully trying to combine existing hypotheses. During this time, it perceived the features "stable," "triangle," "rect," and "unstable" on various objects, leading to a new Hypothesis 4 postulating unstable objects on the right side. Notably, it also perceived the number of objects in a scene at one point, leading to Hypothesis 5 "only in the right scenes, the number of objects is 2." This is a correct solution, although PATHS had no way to know this at that time. Hypotheses 3 and 4 were checked on the new scenes and found to match on some but failed to match on later scene pairs.

Scene Pairs 1–8

During the last part of the run, PATHS iterated through scene pairs one to eight for a second time. It perceived features such as "circle," "triangle," "gets-hit," "count = 1," "position-top," "touching," "collides," "far," and "beside," which led to a number of new hypotheses. An example of a more elaborate one is based on the selector "objects that are beside a circle object at the end." Most importantly though, the algorithm continued to check Hypothesis 5, "only in the right scenes, the number of objects is 2," on each of the scene pairs so that after checking it on scene pair eight, it had been verified on all the scenes of the PBP. At this point, the algorithm stopped the search and reported the solution. Figure 4 shows all hypotheses that were created during the run.

How Path Perceives Physical Scenes

Perceiving the world around us is an active exploration process. People do not attend to every aspect of a scene equally, but instead, pick out specific aspects through constant shifts of attention. In the case of PBPs, once a potential full or partial hypothesis is generated for a rule that distinguishes the left- and right-side scenes, the subsequent descriptions of the scenes will be heavily shaped by that hypothesis. These descriptions involve both the computing of features of the scenes and relations among these extracted features. These two aspects are described in the next sections.

Feature Descriptions

PATHS takes SVG images as its input, which means that each scene is composed of prearticulated objects and a static background with known positions and outlines.

Figure 4
All Hypotheses That PATHS Generated During a Particular Run on PBP02

Hypotheses

R	E	(objects that are 2)	16	1.00
R	E	(circle objects)	4	0.00
R	E	(objects that are far)	2	0.00
L	E	(objects that are 1)	2	0.00
R	E	(objects that are collides-with (rect objects))	2	0.00
R	E	(objects that are beside (circle objects) at the end)	2	0.00
LR	E	(rect objects)	6	0.00
LR	E	(any object)	2	0.00
--	E	(objects that are close (any object))	6	0.00
--	E	(unstable objects)	6	0.00
--	E	(moves objects)	6	0.00

Note. The left-most column states whether a hypothesis matched scenes on the left (L), on the right (R), or on both sides (LR) of the PBP. The next column states that all hypotheses are "exists," rather than "for all" or "for exactly one" quantified. The last two columns show the number of scenes for which each hypothesis has been confirmed and the estimated utility of each hypothesis. The first hypothesis was checked on all 16 scenes and was thus seen to be the solution, so its utility is 1. PATHS = perceiving and testing hypotheses on structures; L = left scene hypothesis; R = right scene hypothesis; E = existential quantifier; PBP = physical Bongard problem. See the online article for the color version of this figure.

The model currently has 34 built-in feature detectors, including detectors for static object properties, such as "size" and "shape," physical properties, such as "stable" and "moves," spatial relations, such as "left-of" or "close," and group attributes, such as "object count." Each feature detector can perceive its respective feature on any object or group, and the resulting percept contains the perceived value of the feature as a fuzzy membership degree (cf. below) between 0 and 1. The default threshold (0.5) to decide whether a feature is considered active or not can be adjusted in each feature.

While perceiving the scenes and searching for a fitting interpretation, more complex features are constructed based on this basic set of features. When the PATHS model works on a PBP, it starts off without any knowledge about the objects in the scenes. Only by actively selecting a feature and a target object and perceiving the feature on the target does the model gradually build internal representations of the scenes. Each feature is associated with a procedure to perceive it using the input description for each of the PBP scenes.

Simulation to Support Perception

In addition to its built-in static and geometric features, PATHS also computes features of scenes assuming that the objects are physical entities. While a major approach to model thinking about physical situations has been to represent them symbolically and use logical inferences to reason about them (Forbus, 1994), other researchers have explored the idea of physical reasoning as running mental simulations. The mental simulations do not require abstract knowledge of physical laws, instead relying on a practical understanding of how one situation turns into the next. They are cognitively plausible (Barsalou, 1999) and fit experimental results with humans better than several rule-based models, for example, when making predictions about the stability and falling patterns of Jenga towers (Battaglia et al., 2013).

PATHS uses a 2D physics engine to model mental physics simulations. A physics engine is a computer program that numerically simulates physical scenes to predict a scene's future states. While very roughly psychologically plausible (T. D. Ullman et al., 2017), we are not intending to make a strong commitment to people's internal simulations being either precise or unbiased. In fact, future iterations of PATHS should incorporate known discrepancies between people's naive physical models and natural physics (Ludwin-Peery et al., 2020; McCloskey & Kohl, 1983).

When provided with the positions and outlines of all objects in a scene, the physics engine in PATHS can simulate the unfolding of events to provide the following object properties for each object at a given time in the future: position and velocity, distance to other objects, and collisions between objects that have occurred. All features that are derived from the simulation data are represented as values of fuzzy membership functions for percepts, expressing the perceived value of the feature as a membership degree or satisfaction value from the range 0 to 1. The distance between objects is captured by the concepts *touch*, *close*, and *far*. Other physical concepts, such as *moves*, *stability*, *supports*, and *movability* require additional derivation.

The *moves* attribute captures whether an object moves or is about to move at a particular point in time. In order to find out whether an object is about to move, the model triggers a short simulation of the situation assuming that all objects start at rest and checks whether the object in question moves in the immediate future.

In the context of dynamic rigid objects, a natural operationalization of object stability is its ability to withstand external forces without moving. If an object can tolerate strong perturbations relative to its mass without toppling over or rolling away, then it is relatively stable. To assess stability, the physics engine is first used to predict the near future of the scene without any external perturbations. If the target object significantly moves, it is considered unstable. Otherwise, the algorithms reset the scene and conduct a series of three short simulations, with increasingly strong horizontal impulses applied to the target object's center of mass at the start of the simulation. If the target object topples over, falls down, or rolls away—all significant changes of position and orientation—the object is considered unstable. The perceived stability has a membership value between 0 and 1 that reflects how strong an impulse was needed to push the object out of balance.

The concept of *support* is closely related to stability in that it describes whether the presence of an object helps to stabilize another object. One aspect of support with separate, rigid, and nonbreakable objects reflects whether the two objects touch. We will call the support relation direct if they do and indirectly if they do not. A second aspect reflects the redundancy of the support an object provides. An object might be the only supporter of another object or it might be part of a group of supporters. Relatedly, a third aspect of support is whether the supported object would actually fall or topple over without the supporter or remain stationary but become less stable. In order to determine whether an object supports another object, PATHS uses the same counterfactual reasoning as with perceiving stability. The model "imagines" what would happen when the supporter is removed by running the respective physical simulation. If the potentially supported object starts to move or becomes unstable after the removal of the supporter, but not if the supporter remains, then a supporting relation between the objects is inferred. The model perceives four different types of support: direct, indirect, stabilizing, and no support and currently maps them to gradual memberships to a single "supports" concept.

An object's *movability* reflects whether it can be moved by a moderate force in different directions. We implemented the perception of movability by letting the model run simulations in which it continuously pulls on a virtual string that is attached to the center of the object in question. The change, or lack thereof, in the object's position when pulling moderately on the string is used to judge its movability. The main reason for a lack of movability of objects in PBPs is that the path of the object might be blocked by other objects or the ground.

Spatial Relations

The solution to many BPs involves not just simple attributes of objects such as *small* or *square* but relations between objects. Many cognitive psychologists have pointed out the importance of representations that go beyond attributes or even conjunctions of attributes, by explicitly representing relations between objects (Gentner, 1983; Loewenstein & Gentner, 2005; Markman & Gentner, 1993; Medin et al., 1993). In PATHS, the relative position of a target object A in relation to a reference object R can be described in terms of distance and direction and will depend on the objects' shapes, especially if they are close to each other.

PATHS can perceive differences in the degree of closeness of two objects, such as very close versus close. This makes it possible either to use the degree of closeness in a categorization rule directly or to adjust a threshold value used to make a binary decision about closeness dependent on a PBP's context—the other scenes in a PBP. To this end, we adapt Isabelle Bloch and colleagues' framework for representing fuzzy spatial relations (Bloch, 1999; Hudelot et al., 2008). PATHS attaches an activation value between 0 and 1 to each perceived spatial relation, reflecting how well the spatial relation fits the situation. The computed spatial concepts can be used to answer two different questions:

1. To what degree does a given spatial relation hold for two specific related objects, for example, is *A* left of *R*?
2. At which locations is a given spatial relation fulfilled for a specific reference object, for example, which places in space are left of *R*, and to what extent?

The answer to the second question is provided by calculating a fuzzy set, referred to as fuzzy landscape, around the reference object in the same image space that the target object is in. The fuzzy set is a function $\mu: \boldsymbol{S} \rightarrow [0, 1]$, which maps each point in the image plane $\boldsymbol{S}$ onto a membership value between 0 and 1. This membership value corresponds to the satisfaction of the spatial relation in question. Figure 5 shows the fuzzy landscapes of the six basic spatial relations using a rectangle as the reference object.

To answer the first question, the fuzzy landscape around *R* is compared to the object *A*. The degree of relationship membership between *A* and *R* is measured by the relationship satisfaction values at the positions in the landscape that are covered by *A*. Bloch (1999) uses three values to represent the fuzzy relation between *A* and *R*: the minimum satisfaction value Π, the mean satisfaction value *M*, and the maximum satisfaction value *N* over all points of *A* in *R*.

$$\Pi_{\alpha}^{R} = \min_{x \in A} \mu_{R,\alpha}(x), \tag{1}$$

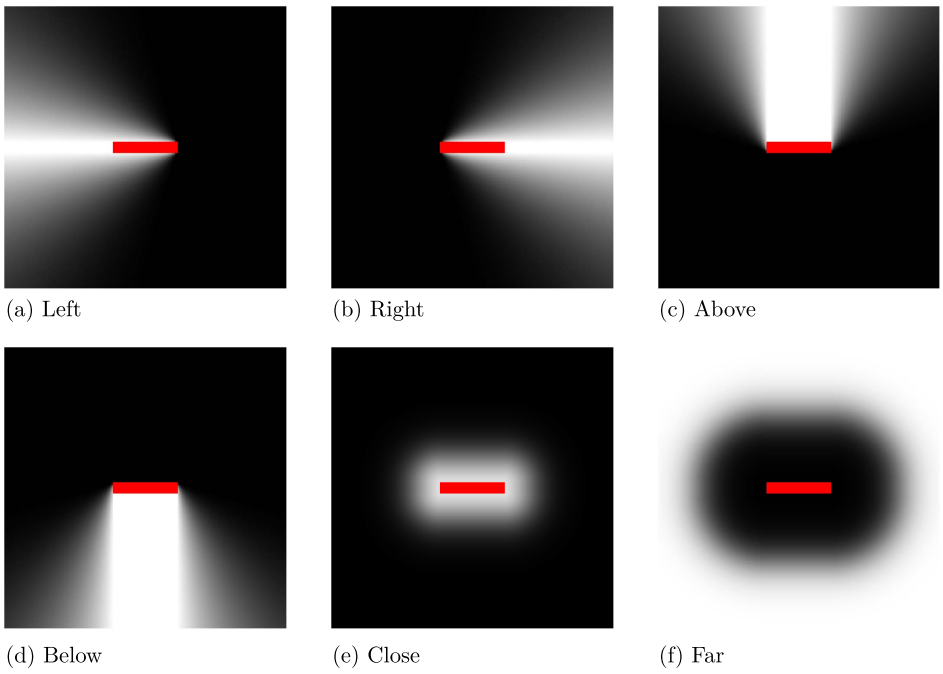
$$M_{\alpha}^{R} = \frac{1}{|A|} \sum_{x \in A} \mu_{R,\alpha}(x), \tag{2}$$

$$N_{\alpha}^{R} = \max_{x \in A} \mu_{R,\alpha}(x). \tag{3}$$

In the fuzzy set framework, the minimum and maximum satisfaction measures can be interpreted as the necessity and possibility of *A* and *R* being separated along relative direction α, respectively. Two advantages of this approach are its mathematical grounding in fuzzy sets theory and fuzzy morphological operators and its flexibility. The relative separation of two objects along a direction at a chosen angle can be calculated efficiently in 2D and 3D to measure concepts such as "left" and "above," whereas morphological structuring elements can be designed for other relations such as the distances "close" and "far." By using the fuzzy *t*-norm and *t*-conorm, conjunctions and disjunctions of spatial maps can be easily calculated in order to construct combinations of spatial concepts such as "far above" and "beside."

A fast algorithm for calculating fuzzy landscapes works in two stages: finding reference points and computing acceptability values. The goal of the first stage is to find the point *Q* in the reference object *R* that is best aligned with the target direction seen from each point

Figure 5
The Fuzzy Landscapes of Six Basic Spatial Relations for a Rectangle



Note. The brightness reflects the degree to which a position in the image space satisfies the respective relation to the rectangle. See the online article for the color version of this figure.

P in the image space $\boldsymbol{S}$. First, each pixel in the image space that overlaps with the reference object is assigned its own position as initial reference point. Next, the algorithm propagates reference points from neighboring pixels, first in a forward direction (left to right, top to bottom) and then in a backward direction (right to left, bottom to top). During each propagation pass, each pixel is set to the best reference point of its eight neighboring pixels (or its own reference point, in case it is already the best). The goodness of reference points for the different spatial relations β is calculated using two different functions β_α, β_{dist}: $\mathbb{R}^2 \rightarrow \mathbb{R}$ that map the relative position of a reference point $Q \in R$ and a point P in $\boldsymbol{S}$ onto a real value. The value expresses how strongly the relative position of both points matches the tested spatial relation β. For direction relations, $\beta = \beta_\alpha$ is used,

$$\beta_\alpha(P, Q) = \arccos \frac{\overrightarrow{QP} \cdot \vec{u}_\alpha}{\left\| \overrightarrow{QP} \right\|}, \tag{4}$$

where $\vec{u}_\alpha$ is the unit vector in the respective direction. For the relations "close" and "far," the second function β_{dist} that evaluates Euclidean distance is used:

$$\beta_{dist}(P, Q) = \left\| \overrightarrow{QP} \right\|. \tag{5}$$

After iterating over the image two times, each point $P \in \boldsymbol{S}$ has an (approximately) optimal reference point $Q \in R$ attached and the respective value x of $\beta(P, Q)$. In the second stage of the algorithm, this value is mapped onto an acceptability value between 0 and 1 by using the appropriate mapping for the relative directions and distances: f_α for direction, f_{close} for nearness, and f_{far} for farness

$$f_\alpha(x) = \max\left(0, \left(1 - \frac{2\|x\|}{\pi}\right)^3\right), \tag{6}$$

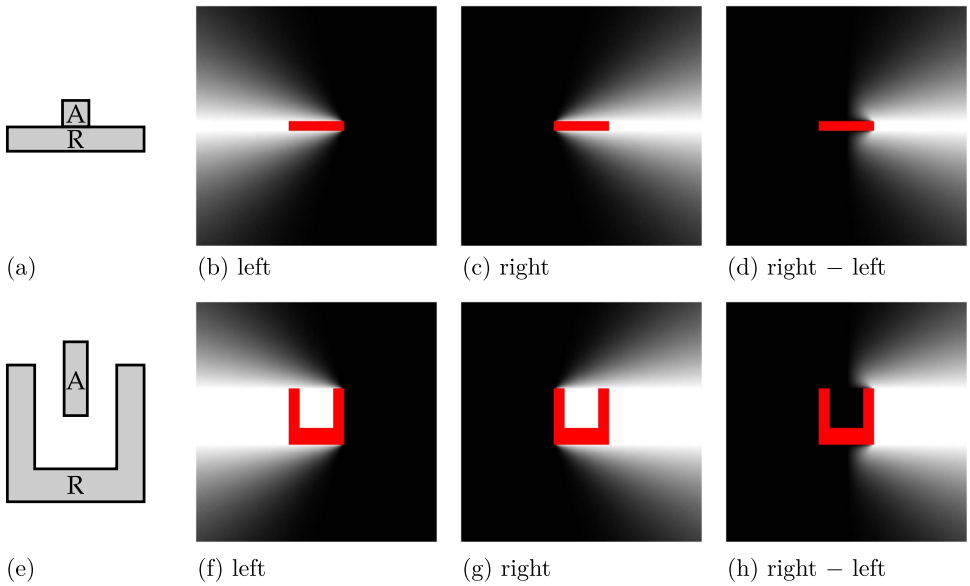
$$f_{close}(x) = 1 - \frac{1}{1 + e^{a(b-x)}}, \tag{7}$$

$$f_{far}(x) = \frac{1}{1 + e^{a(c-x)}}. \tag{8}$$

Bloch (2010) describes how to extend the fuzzy spatial framework to include bipolar information. The notion of having a fuzzy set μ: $\boldsymbol{S} \rightarrow [0, 1]$ is extended to a bipolar fuzzy set, which is a pair of membership functions (μ, ν). The first membership function μ denotes the positive information of where a spatial relation is known to be satisfied. The second membership function ν denotes the negative information of where a spatial relation is known to be unsatisfied. In the context of PBPs, people often consider spatial relations to be in opposition, such as left and right, above and below, as well as near and far. To come up with a psychologically intuitive acceptability value for a relation between two objects, the negative and positive information of the respective bipolar fuzzy landscapes have to be combined. This is done by subtracting the negative landscape from the positive landscape at each corresponding point in the fuzzy set and clipping all values below zero (see Figure 6, for two examples). Notice that without taking the negative information into account, both examples would lead to a high acceptability of both "A left of B" and "A right of B," which would be counterintuitive.

This bipolar nature of many spatial relations is captured well by the bipolar fuzzy set theory. Taking the conjunctive fusion of two bipolar fuzzy sets (μ_1, ν_1) and (μ_2, ν_2) is done by taking the conjunction of the positive parts μ_1 and μ_2 and the disjunction of

Figure 6
Two Example Situations of an Object a Placed Relative to an Object R



Note. If only the fuzzy landscape of the relation *right of* were considered, object *A* would be described as *right of R* in both cases, which is counterintuitive and too permissive. In a bipolar interpretation of the relation *right of*, the opposing relation *left of* is taken into account as negative information. This yields a more intuitive representation. See the online article for the color version of this figure.

the negative parts v_1 and v_2 as negative evidence. The disjunctive fusion of two bipolar fuzzy sets is defined as the disjunction of the positive information and the conjunction of the negative information.

$$(\mu_{\text{and}}, \nu_{\text{and}}) = (\mu_1 \cap \mu_2, \nu_1 \cup \nu_2), \tag{9}$$

$$(\mu_{\text{or}}, \nu_{\text{or}}) = (\mu_1 \cup \mu_2, \nu_1 \cap \nu_2). \tag{10}$$

The conjunction $\mu_\cap$ and disjunction $\mu_\cup$ of two fuzzy sets μ_1 and μ_2 (which could represent either positive or negative information) are defined as:

$$\mu_\cap(i,j) = T(\mu_1(i,j), \mu_2(i,j)), \tag{11}$$

$$\mu_\cup(i,j) = \perp(\mu_1(i,j), \mu_2(i,j)). \tag{12}$$

where T is the fuzzy t-norm and $\perp$ is the fuzzy t-conorm. In the literature on fuzzy mathematics, there are several well-established fuzzy logic systems with their respective t-norms and t-conorms. Three of the most common ones are the Łukasiewicz logic (Łukasiewicz t-norm and bounded sum t-conorm), the Gödel logic (minimum t-norm and maximum t-conorm), and the product logic (product t-norm and probabilistic sum). PATHS uses the minimum t-norm and maximum t-conorm.

$$T_{\min}(a,b) = \min(a,b), \tag{13}$$

$$\perp_{\max}(a,b) = \max(a,b). \tag{14}$$

Positive and negative information of the bipolar fuzzy landscape is calculated using the function

$$\text{sub}(a,b) = \max(0, a-b). \tag{15}$$

These formulas above also allow for the representations of combined spatial concepts. The combined spatial concept *left and close* can be formed as a bipolar fusion of the concepts *left* and *close*:

$$(\mu_{lc}, \nu_{lc}) = (\mu_l \cap \mu_c, \nu_l \cup \nu_c). \tag{16}$$

were the indices *lc*, *l*, and *c* stand for *left and close*, *left*, and *close*, respectively. The negative information for *left*, v_l, and *close*, v_c, is identical to the *right* and *far* relations, respectively. Likewise, the concept *beside* is introduced as a disjunctive fusion of the *right* and *left* concepts.

We define the concept *inside* as conjunctive fusion of *left* and *right*. The intuitive understanding of *inside* is closely related to the convex hull of the reference object. The concept of A being on top of B is modeled as A being above B and touching B. These fuzzy logic descriptions do not fully capture the rich semantic meanings of *beside*, *inside*, or *on top of* (Regier, 1996; Regier & Carlson, 2001) but do provide a concrete and efficiently computed foundation for computing several of the many spatial relations needed to solve PBPs.

How PATHS Builds Scene Descriptions

Basic and Derived Descriptions

One of the interesting aspects of PBPs is that the concepts that are immediately available to a person looking at the scenes are often just the building blocks for more complex features that are needed to formulate a solution. PATHS implements several ways of deriving new features. First, PATHS simulates what will happen in a scene, taking potential interactions among objects into account. An exhaustive search over all possible interactions and temporal relations is infeasible, so choices about what descriptions to form must be made in a context-dependent manner. Second, basic descriptions can be combined to arrive at higher level descriptions. For example, by perceiving the basic spatial relations left and right, a notion of "beside-ness" of two objects can be constructed as their disjunction. Third, forming groups of objects that belong together provides a convenient abstraction over a set of objects and reduces a scene's complexity. Grouping can be based on the similarity of a set of objects along any feature dimensions or their spatial proximity. Fourth, turning initially metric features like the distance between two objects into a qualitative concept with a degree of membership provides an intermediate level of abstraction that is often used by humans and is well suited to construct solutions for PBPs.

The Core Loop of PATHS

In each iteration of PATHS' core loop, the model performs the following steps. First, it probabilistically selects the type of action to perform next from a set of three types. Second, it performs the action, which might involve the stochastic selection of action parameters. Third, it performs potential follow-up actions. Fourth, it probabilistically decides whether to shift attention to a new set of scenes. Figure 7 shows an overview of this architecture, which is decomposed below.

Switching Between Active Scenes

The influence of the order in which the scenes are perceived can be studied for both PATHS and people. To that end, two scenes of a PBP are presented at a time, to simulate comparison processes known to be common for people (Forbus et al., 2017; Goldstone et al., 2010; Rittle-Johnson et al., 2020). The sequence of scene pairs is fixed during a particular trial, while the decision of when to uncover the next scene pair is up to the human or cognitive model, consistent with human experimental paradigm to be described later.

For PATHS, this decision is based on how promising the current hypotheses are. If there is a promising solution candidate that has already been checked on the current scenes, the model is more likely to move on to the next scenes earlier so the solution candidate can be further verified. If none of the hypotheses are particularly promising, or if a promising one yet has to be checked on the current scenes, the model is likely to continue looking at the current scenes.

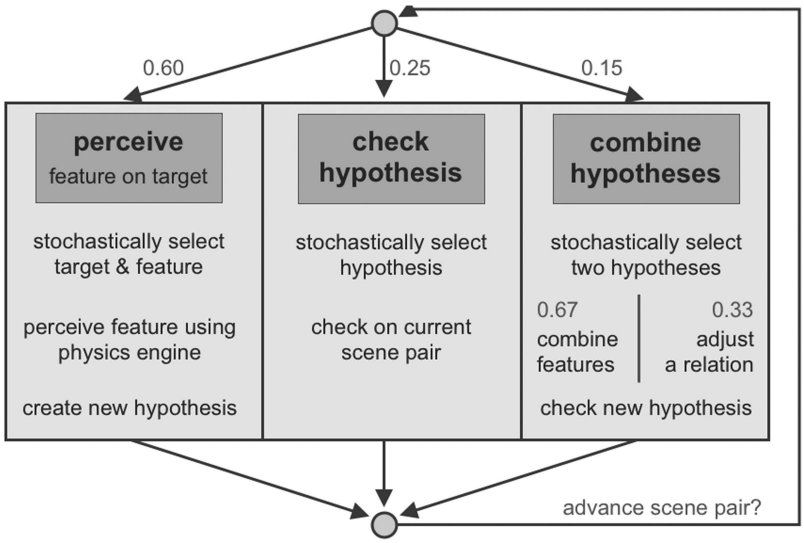
Objects and Groups

PATHS keeps track of all descriptions performed on an object. Groups are constructed by selectors that select a subset of objects in a scene, such as "square objects" or "any object." Each group of objects can be a target for perceiving group features like "object count" and will keep track of all selectors that are known to select that subset of objects in the group.

Selectors

Selectors abstract from observations on a specific object or group and represent a structured description. The abstraction is both from

Figure 7
PATHS' Main Loop



Note. On each execution step, PATHS probabilistically selects one of three action types "perceive," "check hypothesis," and "combine hypotheses" with the indicated Probabilities. The "combine hypotheses" block in the diagram has two subtypes, of which one is selected according to the specified probabilities. Some actions trigger the creation of a new hypothesis, which in turn triggers a check hypothesis action on the created hypothesis. PATHS = perceiving and testing hypotheses on structures.

specific objects to descriptions of those objects and from graded to discrete feature values. The percept of a feature like "small" is represented with a membership value in [0, 1]. When turning that percept into a *feature matcher*, the value is discretized into a binary value "true" or "false," so that the resulting selector can either match objects that are "small" or "not small." The default threshold of 0.5 can be adjusted per selector and feature type to account for the observed distributions of feature values in the left and right PBP scenes.

When a selector is applied to a scene, it selects a subset of the scene's objects. Selectors are conjunctions of three kinds of feature matchers: object attribute matchers, object relation matchers, and group attribute matchers. Each feature matcher maps a source set of objects to a target set of objects that contains all those objects from the source set that have the same feature–value combination as the matcher. For example, a selector that contains a single object attribute matcher "small = true" will select all objects in a scene for which the feature "small" has a membership value above its threshold.

An object relation matcher contains, in addition to the relation type and value, a selector for the reference object of the relation. For example, "hits (small)" would select all objects in the scene that hit a small object. Reference selectors are not allowed to contain relation matchers, which prevents complicated nested structures "a circle that is left to a square that is left to a triangle" but allows for "a circle that is left to a square and is left to a triangle."

When PATHS applies a group attribute matcher to a set of objects, it returns the original set if the group of objects in the set has the same feature–value combination as the matcher. Otherwise, it returns an empty set. For example, the matcher "count = 3" will return all objects when applied to a scene with three objects or no objects if the scene has a different number of objects in it.

All matcher types can be easily combined. For example, a selector consisting of the two matchers "close-to(square) ∧ count = 3" will check whether the number of objects in a scene that is close to a square is three. If so, it selects those three objects; otherwise, it selects none.

Hypotheses

A hypothesis represents a potential solution or partial solution to the current PBP. It consists of a selector, the side of the PBP the selector is describing (left, right, or both), and the quantifier that is used with the selector (exists, all, or unique). For example, the "small" selector could be applied as "in the left scenes, all objects are small," or as "in the right scenes, there is a small object," or as "in all scenes, there is exactly one small object." Each hypothesis keeps track of whether or not the scenes it has been tested on matched and whether all, some, or exactly one of the objects in each scene matched. This information is used to pick the best fitting side and quantifier for describing the scenes seen so far.

Attention Mechanisms

The guiding of attention in PATHS toward regions and aspects of the scenes that are most relevant to the learning process works on several levels. Attention can be shifted to certain objects in a scene, to certain features, or to the aspects necessary for checking a solution hypothesis. In PATHS, attention is modeled as the probability distributions used to select objects and features in PATHS

perception actions. These distributions are continuously adjusted while PATHS works on a problem in several ways. During the initial exploration of the PBP scenes, objects differ in their saliency based on their features. Attention tends to be drawn to objects that are about to move or are unstable, objects that are spatially separated, or objects that are "oddballs" along a feature value. Generated hypotheses influence the choice of what to perceive next in three ways. First, during the exploration of the scenes, PATHS is more likely to attend to the objects that play a role in promising solution hypotheses. Second, PATHS directly checks existing hypotheses on new scenes, and during this process, only perceives what is necessary to confirm or refute the hypotheses. Third, existing hypotheses can be combined, and the resulting hypotheses influence perception as described in the previous two points.

Actions

All work done by PATHS while solving a PBP is organized into small, separate chunks, called actions. A practical way of measuring the model's performance that abstracts from specific computer hardware is to count the number of actions that PATHS performs to solve a PBP. There are three action types that the model can perform, which are: (a) to perceive and potentially create a hypothesis from a perception, (b) to check a hypothesis against the current scenes, and (c) to combine two existing hypotheses to form a new one. These actions are triggered top-down by sampling from the fixed multinomial distribution, as shown in Figure 7.

The *perception* action uses one of two strategies with equal probability. Either the action first selects a target from one of the current scenes and then selects which new feature it should perceive on it, or it first selects a feature and then selects a new target object for perceiving the feature. The former strategy corresponds to a person looking at an interesting target in a scene and perceiving new properties of it, whereas the latter corresponds to a person looking for targets that have a particularly interesting feature. In both cases, the target can be an object or a group of objects, whereas the feature can be a group attribute, object attribute, or a relation between two objects. If an object relation is to be perceived, a reference object for the relation is chosen in addition to the target object. For features that can change over time, such as an object's position in the scene, the perception action stochastically picks whether the feature is perceived in the initial or final situation based on a fixed multinomial distribution ($p = .67$ and $p = .33$, respectively). When perceiving a feature on a target, the model only selects feature–target combinations that were not *actively* noticed before. If something new is perceived during the perception action, the creation of a hypothesis is triggered to turn the percept into a corresponding hypothesis in the next step.

When creating a new hypothesis, PATHS turns the percept into a pattern description—a "selector"—that can be applied to new scenes. Each selector is associated with one hypothesis—a potential solution to the PBP that keeps track of all matching results. To turn a perceived relation into a selector that can be applied to new scenes, the reference object, itself, needs to be represented via a selector. Either the "any object" selector or some more specific selector among the existing compatible hypotheses in the workspace might be picked. If the percept contains a group attribute, a selector that matches this group is picked and combined with the main selector, which describes the perceived group attribute's type and value. After the new selector and its associated hypothesis are created, they are added to the workspace unless an identical selector already exists. In either case, a check-hypothesis action is triggered for the hypothesis. For the purpose of counting the number of actions PATHS uses, the creation of a hypothesis counts as a separate action.

The *check hypothesis* action applies an existing hypothesis to the scenes in the current scene pair and keeps track of the results. Compatible match results will contribute positively to the estimated potential of the hypothesis, whereas incompatible match results will have the opposite effect. In addition to whether the hypothesis matched the scenes or not, the model keeps track of whether exactly one, or a few, or all of the objects in the scenes match the hypothesis' selector. This information is used to decide on the best logic quantifier ("unique," "exists," or "all") to use in the hypothesis. Each time, after checking a hypothesis on a new pair of scenes, the action will pick the side and quantifier for the hypothesis that best fits all previous matching results. For example, "in the left scenes, there is a small object" could be changed to "in the right scenes, all objects are small." The goal of these adjustments is to find a hypothesis that only matches scenes from one side and is therefore a potential solution. If that is not the case anymore after checking a scene pair, the check-hypothesis action attempts to "repair" the hypothesis by readjusting the concept-membership thresholds of the selector's feature matchers. For example, the selector might be adjusted to accept a larger range of object sizes as being "small."

If the *combine hypotheses* action is selected by the model, the model will further choose stochastically among two subtypes: directly combining two hypotheses or using one hypothesis to modify the relationship in another hypothesis. In the first case, the action probabilistically selects an object from one of the active scenes and picks two hypotheses that match the selected object and have not been combined before. Only hypotheses that match scenes from both sides are considered because combining hypotheses always results in a more specific hypothesis, and hypotheses that have so far matched scenes from only one side are still sufficiently specific. A check-hypothesis action is triggered for the hypothesis created by the conjunction of the two former hypotheses. The reason to select two hypotheses through a common object they select is to ensure that they are not incompatible with each other—their conjunction selects at least one object in one of the scenes. This roughly corresponds to a person noticing that the same object is both small and stable, so in addition to the solutions "there is a small object" and "there is a stable object," the observer might now consider the solution "there is a small and stable object."

If the relationship subtype of the *combine hypotheses* action is selected, PATHS will create a new hypothesis by replacing the reference object selector of a relation feature in one hypothesis with a different selector. For example, based on the hypothesis "there is a square on top of an object," a new hypothesis "there is a square on top of a big object" can be created if there is an existing "big objects" selector. Specifically, the combine-hypothesis action first stochastically selects a hypothesis that has a relation feature and one of the objects in the current scene that matches the relation's reference object selector. Then, the action searches for a different selector selecting this reference object and, if successful, creates a new hypothesis that is a copy of the original one with the relation reference selector replaced by the new selector. Finally, a check-hypothesis action is triggered for the newly created hypothesis.

In some cases, an action may not be successfully completed by PATHS. For example, after combining two hypotheses to create a new one, PATHS may find that an identical hypothesis already exists. In those cases, the unsuccessful action is still counted toward the number of actions that PATHS took to find a solution.

Comparison of PATHS to Humans

PATHS was designed to be a plausible cognitive model of open-ended rule-based concept learning in humans applicable to situations involving both ongoing perceptual encoding and symbolic hypothesis testing. In comparing PATHS to human solutions of PBPs, we consider overall performance on different problems, the time required to generate solutions, and the influence of different presentation conditions. The performance of PATHS is evaluated by repeatedly running it on PBPs presented to people across several experiments (Weitnauer, 2016; Weitnauer et al., 2013, 2014). The proxy that we chose for response time in PATHS is the number of actions that is required, on average, by PATHS to solve a problem. Different actions in PATHS could involve cognitive operations that vary in complexity and therefore require different amounts of time. A more nuanced analysis would assign empirically determined times to different actions, but this refinement was omitted from the current analysis because efforts were taken to create interpretive actions that might be expected to correspond to single human actions (Anderson, 2009). Another simplifying assumption is letting the PATHS model accurately remember which hypotheses it has already checked and discarded (visualized in four as grayed entries in the hypotheses list). On the other hand, humans will often reconsider previously discarded hypotheses (Bruner et al., 1977)

Across these previously reported experiments, participants were shown a subset of the PBPs shown in Appendix B and were asked to supply a solution (for a table of solutions of the PBPs in Appendix B please see Appendix C). To avoid conflation of solution time with time to formulate the solution in written form, participants were asked to perform a mouse click at the moment they felt they had found a solution, which was immediately typed in English into a text box, and subsequently assessed by human judges to determine whether it would perfectly distinguish the left and right scenes. In the standard, simultaneous display version of this task, participants saw all of the scenes that comprised a PBP at the same time, and these scenes remained on the screen until either the participant guessed the rule or gave up.

The average correct response times of human participants and PATHS on each problem were compared. The human response time data contained outliers. Also, the response times on unsuccessful trials are difficult to interpret and result from very different mechanisms in PATHS and the human participants. PATHS always searches for a solution until a fixed maximum number of actions, whereas humans may give up or submit a wrong answer at any time. To address these issues, we excluded all trials on which the response time was longer than 10 min. There are no fast response time outliers because the correct answer cannot be guessed.

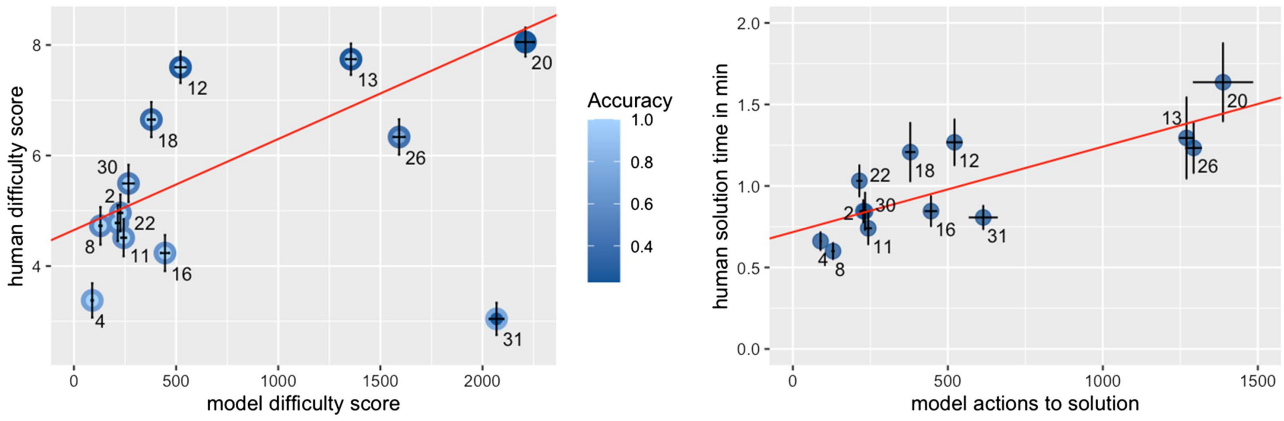
The left panel of Figure 8 compares the relative difficulty scores of the problems for PATHS and people, whereas the right panel shows a scatterplot of human versus model response times only considering solved trials. The difficulty score combines accuracy and reaction time information by setting the reaction time of all unsolved trials to 10 min for humans. Any trials that took longer than 10 min are also set to 10 min. The model data already have all unsolved trials set to a value of 2,500 actions and remains unchanged. More formally, the difficulty score for humans is defined as:

$$\text{difficulty} = \begin{cases} \min(10, \text{RT}), & \text{if solved} \\ 10, & \text{if unsolved} \end{cases} \tag{17}$$

Paralleling the human measure, the model difficulty score takes into account both speed and accuracy by assigning a value of 2,500

Figure 8

Comparisons of PATHS to Human Performance



Note. The left panel plots the difficulty score for humans (solution time in minutes, capped at 10 min) versus the model (solution time in actions, capped at 2,500 actions), averaged per problem. For unsolved trials, the difficulty score is set to the cap values (10 min or 2,500 actions, respectively). The brightness of the inner circles represents the model's successful solution rate, and the outer circles represent the subjects' successful solution rate. The red line shows a linear fit to the data excluding Problem 31. The right panel plots the average response time for humans and the average number of actions by PATHS, only considering solutions for both humans and PATHS that were correct. In both plots, the error bars represent standard errors. PATHS = perceiving and testing hypotheses on structures. See the online article for the color version of this figure.

actions for any problem not solved by PATHS. The Pearson product–moment correlation coefficient for the response times between humans and PATHS is 0.33, with $t = 1.17$, $df = 11$, and p value = .264. There is a single problem, PBP 31, for which the difficulty for humans and the model was very different. If we remove problem 31 from the data, we get a correlation of $r = 0.74$, with $t = 3.4$, $df = 10$, and p value = .006 for the human and PATHS difficulty scores. The relative difficulty of PBP 31 for PATHS compared to humans is likely because humans have richer background knowledge that is relevant. All solutions found by the model involve the "can-move-up" attribute, with the two most common being "there are can-move-up circles in all left scenes" and "there are small and can-move-up objects in all left scenes." Human participants came up with a wide variety of solutions, using verbs that evoke rich situations. They described the circle as being trapped, enclosed, covered, stuck, protected, imprisoned, contained, boxed in, hidden, surrounded, confined, secured, shielded, "having an escape," "can be picked up," "can get out," and free to move. While there is little to guide attention in PATHS toward relevant concepts, humans quickly hone in on a familiar narrative that is captured in the scenes.

Despite this particular difference between humans and PATHS, the overall results show strong parallels between PATHS and humans in terms of the order of difficulty of PBPs, and the time required to solve them. Even more basically, the overall rate with which PATHS solves the problems is comparable to humans, with humans solving about 45% of the problems, whereas PATHS regularly solves 40% of the same problems. Furthermore, these rates of solution for these PBPs are far higher than other computational models for solving BPs (Depeweg et al., 2018; Foundalis, 2006; Yun et al., 2020). To be sure, these other computational models were not designed to solve PBPs, and so the problems that we tested are outside of the scope of these models. Still, the ability to solve BPs involving physical simulations is a competency that people readily demonstrate, and one that is also present in PATHS but not in other computational models for solving BPs.

Sequential Effects on Category Learning

An important way in which PATHS differs from other models for solving BPs is that it processes the scenes sequentially rather than in a batch, and it takes into account which scenes are being considered at the same time. Accordingly, PATHS can potentially accommodate experiments in which the order of scenes that make up a PBP has been manipulated. Our previous studies (Weitnauer et al., 2013, 2014) have varied which scenes participants are likely to consider at the same time in two ways. When all scenes that make up a PBP are presented *simultaneously*, then which pairs of scenes are likely to be considered together is manipulated by presenting them spatially close together, on the same row, of a PBP. In other experiments with *sequential* presentations, we present only one pair of scenes at the same time, either from the same or different sides of the problem. Figure 9 shows how similarities within and between a category/side can be independently manipulated. When within-category similarity is high, then the two scenes from the same category that are spatially (when simultaneously) or temporally (when sequential) adjacent are similar to each other in terms of their overall configuration of elements. This is also true for between-category similarity, but in this case, the scenes that are juxtaposed belong to opposite categories. Between- and within-category similarity is independently manipulated. For example, in the upper right quadrant of Figure 9, the juxtaposed scenes across the categories (i.e., the scenes that occupy the same positions within their categories) are similar to one another, but the juxtaposed scenes (i.e., the scenes that are beside each other) within a category are dissimilar.

As with participants in the sequential condition, each of the PBPs is presented to PATHS as a sequence of scene pairs. PATHS proceeds through a sequence of scene pairs to match the order in which human participants saw the pairs in sequential presentations or the spatial juxtapositions of scenes for the simultaneous presentations. After a perception action has produced a selector for one scene in a pair, the same selector is likely applied to the other paired scene. In this manner, the pairing of scenes has a large influence on the descriptions noticed by PATHS and the hypotheses formed.

Collapsing across whether juxtaposition is temporal or spatial and whether scenes are paired within category or between category, the empirical results are shown in Figure 10. For humans, our primary measure of success is the adequacy of their stated rule for distinguishing the left and right scenes. As a secondary measure, we presented people with novel scenes and measured their percentage of correct categorizations (Weitnauer et al., 2013, 2014). The secondary measure has the advantage of not requiring any human judgments to assess solution quality but is not directly comparable to PATHS because the output for PATHS is a rule. In any case, these two measures are highly correlated for humans. People tend to be able to state the correct rule when and only when they can accurately categorize new scenes.

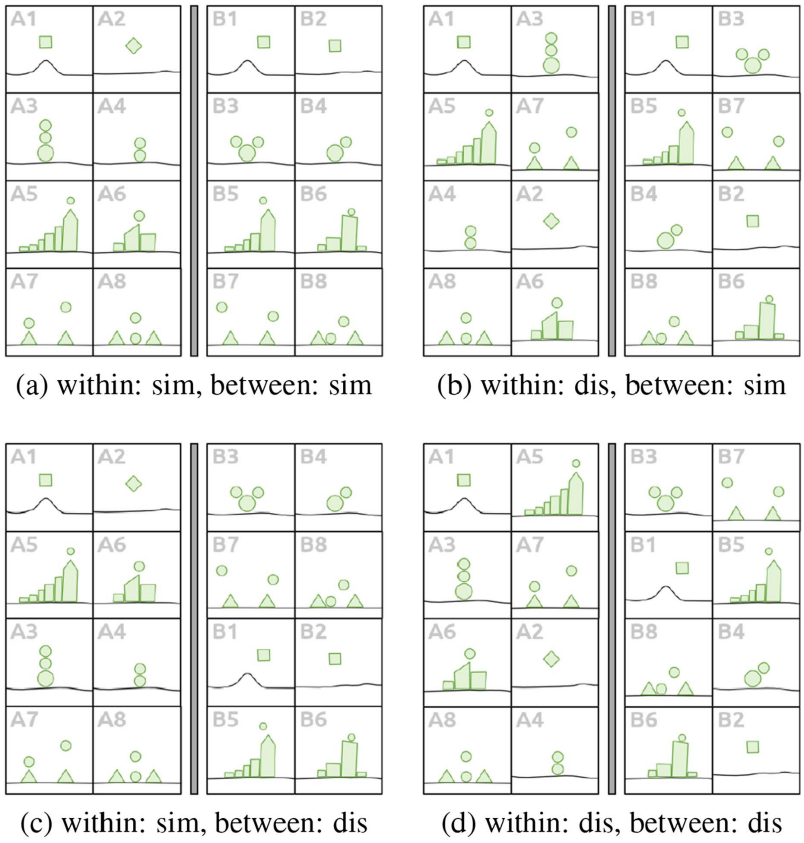
For both humans and PATHS, the difficulty of a problem is measured by both how often it is solved and how long it takes to achieve the solution when it is solved. This difficulty score is a more sensitive measure than simple accuracy because participants vary widely in terms of how long they take on a problem before giving up. The time required to solve PBPs is very heavy tailed for both humans and PATHs, which is why difficulty is log transformed. The similarity within a category has an opposite influence on problem difficulty compared to similarity across categories for both people and PATHS.

When scenes belonging to opposite categories are similar rather than dissimilar to each other, then this makes the problem easier to solve (the log difficulty decreases). This result follows naturally from the notion of discriminative contrast (Carvalho & Goldstone, 2014, 2015; Kang & Pashler, 2012). It states that direct comparison of instances from different categories highlights their differences, together with the insight that comparing similar instances is especially effective because there are fewer superficial differences and the alignment of instances is easier. Also relevant is Winston's (1970) notion of "near misses," which allows an arch concept to be efficiently learned from closely paired positive and negative examples. The same applies to the finding from Markman and Gentner (1993) and Christie and Gentner (2010) that it is easier to structurally align two similar scenes than two very different scenes, and this alignment process promotes noticing their crucial differences.

The opposite influence of similarity on within-category comparisons fits another literature on concept formation. Since comparing instances of the same concept can serve to highlight commonalities

Figure 9

Different Orderings of Scenes Within Problems From Weitnauer et al. (2013, 2014)



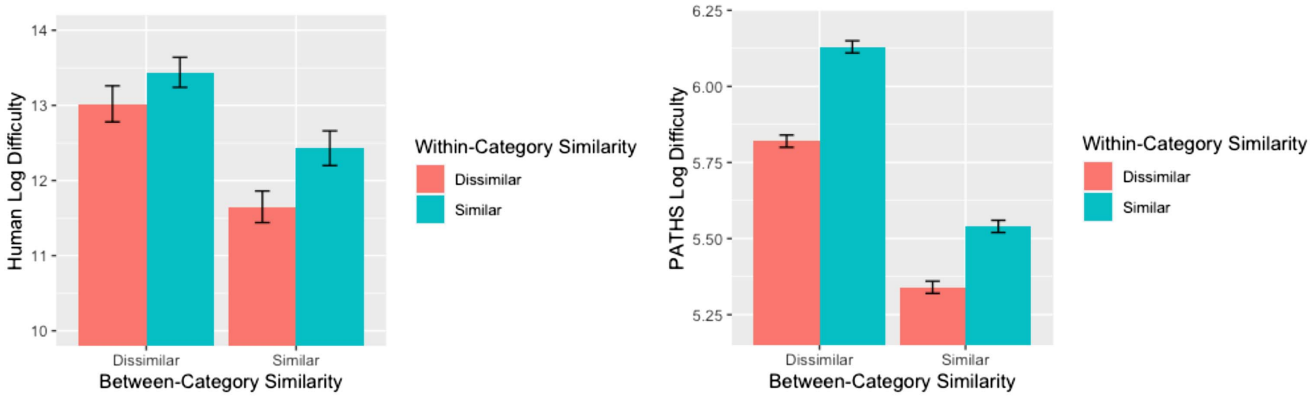
Note. (a) The scenes that are juxtaposed close to one another within a category/side, and also across categories, are similar to one another. (b) The juxtaposed scenes within a category are dissimilar but across categories are similar. (c) The juxtaposed scenes within a category are similar but across categories are dissimilar. (d) The juxtaposed scenes are dissimilar both within and between categories. See the online article for the color version of this figure.

between them, it may be beneficial to compare instances that share as few features that are irrelevant for the characterization of the concept as possible. A closely connected notion, called "conservative generalization" by Medin and Ross (1989), is that people will generalize as minimally as possible, preserving shared details unless there is compelling reason to discard them. As within-category objects become more similar, their superficial similarities might be mistaken as defining ones and might lead to too narrow a category representation. This may occur, for example, when learning to discriminate pairs of similar-sounding words (Rost & McMurray, 2009) or when learning about which methods to use in exploratory data analysis (Chang et al., 2003). By varying the irrelevant features possessed by examples within a single category, the relatively stable, deep commonalities stand out and can make hard learning tasks, like learning relational syntax rules from examples, feasible (Gómez, 2002). Another example of the benefit of low within-category similarity when learning from examples is the results of Halpern et al. (1990), who asked students to read scientific passages that included either "near" (superficially similar) or "far" (superficially dissimilar) analogies. The passages that included far analogies led to superior retention, inference, and transfer compared to those featuring superficially similar comparisons, which showed no benefit at all.

PATHS shows both discriminative contrast and conservative generalization effects by virtue of its tendency to apply the same selector to both juxtaposed scenes. When the juxtaposed scenes belong to different categories, then having high similarity between the scenes helps PATHS quickly identify a selector that applies to only one of the scenes, thus becoming a candidate for being part of the rule that discriminates between the two sides of a PBP. Low between-category similarity leads to many "false alarm" selectors being created that do not successfully apply to other scenes. Conversely, when the juxtaposed scenes belong to the same category, then having low similarity between the scenes helps PATHS quickly identify the selector that they share that is also shared by the rest of the scenes within the category. High similarity between juxtaposed scenes belonging to the same category leads to "false alarm" selectors being created that do not apply to other scenes in the same category. In general, "false alarm" selectors that are created for one pair of scenes but do not generalize to other scenes slow PATHS solutions because the invalid selectors are tested, albeit unsuccessfully, on several scenes once they are generated.

Figure 10

Difficulties of Solving Problems for People and PATHS, as a Function of the Similarities of the Paired Scenes Within and Between Category



Note. Juxtaposing similar scenes promotes rule induction if the scenes come from different categories and hinders rule induction if the scenes come from the same category. Error bars represent standard errors of the mean. PATHS = perceiving and testing hypotheses on structures. See the online article for the color version of this figure.

The empirical and modeling results bring together these two lines of research on the advantages of between-category similarity and disadvantages of within-category similarity for inducing new categories by example. The PATHS model additionally provides a unified mechanism for these converse influences of similarity. The perceptual descriptions that are constructed for a scene are influenced by all of the hypotheses generated while working on a PBP, with an example shown in Figure 4. While the global context of a PBP acts as top-down influence, guiding what is noticed next in a scene, the immediate context of the paired scene has an even greater, more immediate influence on what is noticed next. Newly formed descriptions for a scene are immediately tested on its paired scene. If the paired scene comes from a different category and the description applies to it, then this will immediately lower the priority of the description because it does not, by itself, distinguish between the two categories. Conversely, if the paired scene comes from the same category and the description applies to it, then this will immediately increase the priority of the description because it has now been confirmed for another instance of the same category. In this manner, perceptual descriptions (once established) change the priority of hypotheses, and these same emerging hypotheses guide the prioritization of acquiring additional perceptual descriptions.

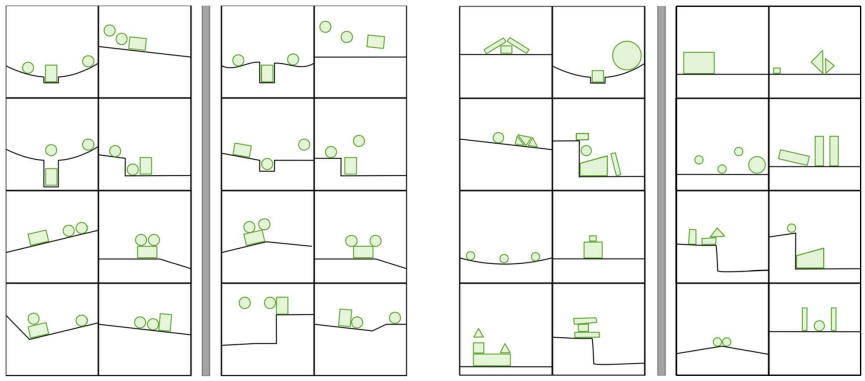
Wtihin- and Between-Category Similarity

While the four PBPs in Figure 9 vary in the similarity of spatially juxtaposed scenes but maintain the same scenes across the categorization problems, other studies have varied the overall similarity of the items being categorized. Previous results have shown that the efficiency of category learning is influenced by the sequencing of items to be categorized. Researchers have often contrasted *interleaving* categories—alternating presentation of items from different categories—with *blocking* categories—presenting many examples of one category followed by many examples from another category. Some researchers have proposed that interleaving is generally superior to blocking because it widely distributes presentations of a category over time (Foster et al., 2019; Kornell & Bjork, 2008) or because it emphasizes features that serve to discriminate between the categories being learned (Kang & Pashler, 2012). Other researchers have argued that whether interleaving or blocking is superior depends on the nature of the categories being learned (Carvalho & Goldstone, 2015; Goldstone, 1996; Zulkiply & Burt, 2013). In particular, when all of the objects across all of the categories are similar to one another, then interleaving is beneficial because it highlights the few, hard-to-find features that discriminate between the categories. Conversely, when all of the objects across all of the categories are dissimilar to one another, then blocking is beneficial because it highlights the few, hard-to-find features that are shared by the members within each category (Brunmair & Richter, 2019; Carvalho & Goldstone, 2014). Carvalho and Goldstone (2022) developed a computational model, sequential attention theory model, that accounts for this interaction between sequence (blocked vs. interleaved) and category structure (high vs. low similarity) by assuming that people place emphasis on features shared by successive items belonging to the same category as well as features that differ between successive items belonging to different categories.

Unlike sequential attention theory model, PATHS does not explicitly apply differential encoding weights to different scene descriptions, but its core processing serves to stochastically interpret the same scene in different ways depending on the other scene with which it is paired. PATHS tends to test descriptions that it has just created for one scene of a pair to see if they apply to the other. This leads to descriptions that are shared by scenes belonging to the same category, and that discriminate between scenes belong to different categories, being likely to be tested on other pairs. Given PATHS's core context dependency, we devised the pair of PBPs shown in Figure 11 to test whether it would demonstrate the same interaction between sequencing (blocked vs. interleaved) and category structure (high vs. low similarity) that has been observed with people. PATHS was run 1,000 times for each version of the PBP with each sequence. When the scenes were presented in a blocked sequence, then two randomly selected scenes from the same category were presented together at one time. In the interleaved sequence, the two paired scenes were selected from opposite categories.

Figure 11

Two Versions of the Same PBP, Varying in the Similarity of the Scenes



(a) High Similarity (b) Low Similarity

Note. Both versions have the same solution: Left scenes end with objects touching and right scenes end with objects apart. PBP = physical Bongard problem. See the online article for the color version of this figure.

The results from the simulations are shown in Figure 12. Consistent with the empirical results described above, there was a highly significant interaction between sequencing and category structure. When scenes were blocked by category, the PBP with dissimilar scenes tended to require fewer steps to solve than the version with similar scenes. Conversely, when scenes belonging to different categories were paired (interleaved sequence), the PBP with similar scenes tended to be solved faster than the version with dissimilar scenes. For both humans and PATHS, interleaving categories is particularly helpful when objects across categories are similar to each other. In that case, there are not many descriptions that discriminate between the objects, so candidate descriptions are likely to be the rule-defining ones. In addition, the simulations show a main effect such that interleaving tends to lead to faster category rule discovery than blocking. This is generally consistent with the literature's frequent overall recommendation to interleave rather than block categories, all else being equal (Doug Rohrer & Hartwig, 2020; Yan & Sana, 2021).

Comparing Alternative Versions of PATHS

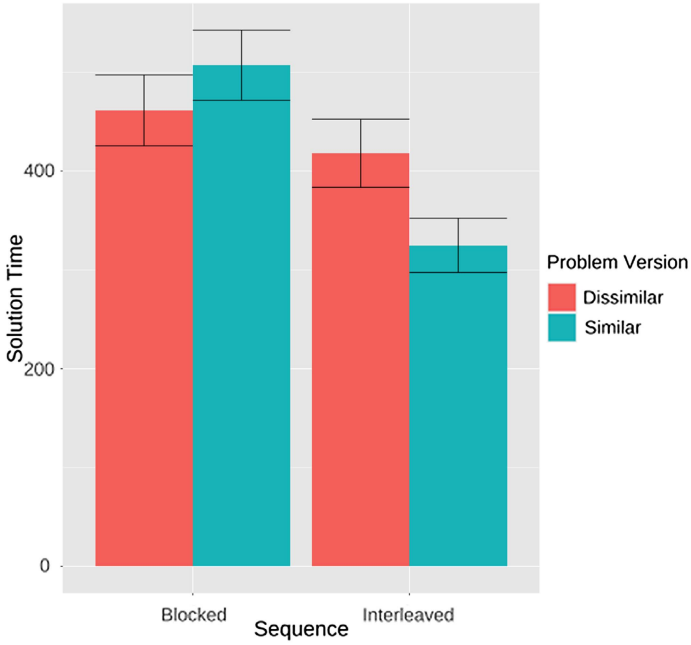
PATHS is a complex model with many interacting components. One way to determine how important a particular component process is for PATHS's successful operation is to modify or lesion that component and observe its impact. This model-lesioning approach has been effectively used in previous models of inductive reasoning (M. Mitchell, 1993). One core component of PATHS is its capability to model physics simulations, which allows it to solve PBPs that rely on dynamic attributes such as the stability or imagined movement of objects. In the first lesioned version of PATHS, we disabled its capacity to run physical simulations. Figure 13 shows a comparison of the no-physics version of PATHS with the original PATHS model. Two observations stand out. First, without the capacity to conduct physical simulations, PATHS can no longer solve any PBPs that rely on dynamic properties, such as problems 8, 12, 18, 20, 22, 26, 30, and 31. Second, the lesioned version of PATHS was able to solve problems that do not rely on dynamic properties faster because it had fewer properties that it was able to perceive and therefore a smaller search space.

Another core component of PATHS is its mechanism to stochastically select among existing solution hypotheses for checking against new scenes and to combine the hypotheses to form more complex hypotheses. Figure 14 compares the original PATHS model with an "exploitation-biased" version of PATHS that strongly focuses on hypotheses with the most matching scenes at the cost of exploring new hypotheses. For easily solved problems, there is no significant difference in the performance of the exploitation-biased variant compared to the original model. For the more complex problems 13, 20, 26, and 31, the exploitation-biased variant performs worse than the original model due to its tendency to extensively check and combine initial hypotheses before considering new, potentially simpler ones. This lesion points to an explore–exploit trade-off (Cohen et al., 2007) in PATHS. Continuing to check hypotheses that have proven to be relevant to a problem is an important way for PATHS to strategically pursue promising trails, but it comes with the cost of limiting broader exploration.

In a final variation, we removed the attention mechanism from PATHS that allows it to adjust how it selects objects for perceiving further attributes. In the original model, PATHS is biased toward perceiving attributes of objects that are featured in promising hypotheses. For example, if a hypothesis with many successfully matched scenes selects rectangles, PATHS is more likely to perceive further attributes on such rectangles and could, for example, notice that a particular rectangle is also small. The lesioned version that does not bias perception toward objects featured in hypotheses has performance similar to the base model, except that it does significantly worse for PBP 26. This makes perfect sense given that PBP 26 has a lot of objects in some of the scenes, and its solution is based on the movement direction of a small circle that is present in each scene. The attention mechanism in the original version of PATHS makes the model more likely to focus on perceiving additional properties of that circle once a hypothesis matches it across all scenes.

The data that we recorded in simulation runs of the model using the different variations is, together with the source code of the

Figure 12
Results From PATHS When Shown the PBPs in Figure 11 in Interleaved Versus Blocked Sequences



Note. Error bars represent standard errors of the mean. Note the relatively large standard error despite a high number of model runs. This is due to the model deciding stochastically what to focus on first, which in turn influences what it perceives and tests subsequently in a solution attempt. Consistent with empirical results using simple visual stimuli (Carvalho & Goldstone, 2014), pairing blocked presentations with dissimilar problems, and interleaved presentations with similar problems, results in relatively faster concept learning than the other two pairings. PATHS = perceiving and testing hypotheses on structures; PBPs = physical Bongard problems. See the online article for the color version of this figure.

model, available in the analysis-new folder at https://github.com/eweitnauer/Dissertation-PATHS-Model.

Comparison of PATHS to Other Bongard Solvers

The challenges connected with BPs are reflected in the still relatively limited number of implemented computational approaches to solve them. This would seem to make comparisons between them an easy matter. However, authors test their systems on different subsets of BPs. They also assume different input representations, which by itself can have a very significant impact on a solver's task. The first computer-based approach (Saito & Nakano, 1995) that was applied to BP problems described the scenes in a BP in terms of logical relations between objects and their attributes and used a search algorithm to identify a formula that could correctly distinguish descriptions for the left and the right sides of a BP. This approach required a manual translation of images into the fixed categories and attributes of their symbolic description. The chosen description space limited such translation to a subset of the original BPs, and the authors found their method able to solve 41 problems in this subset, which is better than any of the subsequent approaches that worked directly on images so as to avoid the manual translation step.

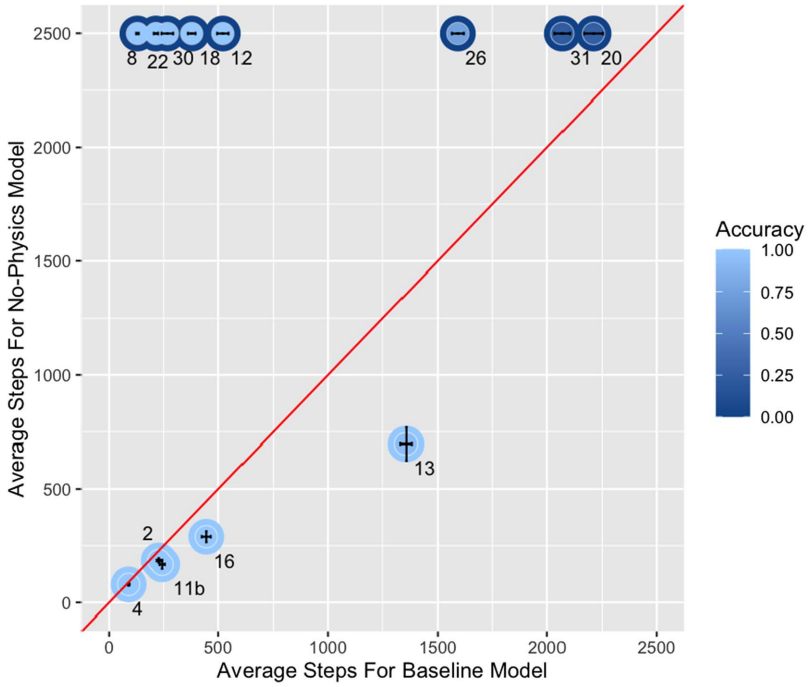
The work by Foundalis (2006) was the first such approach. It worked directly on 100 × 100 pixel images for the scenes, and the development of a visual front end that could handle such images was a significant feat at the time. As a result, this approach offered a more general and autonomous solution to the challenge of concept formation from line drawings. However, Phaeaco's design prevented it from being able to ignore parts of a scene as irrelevant, from representing relational situations like an object being left of another object, and from directly applying an interpretation that it discovered for one scene to another scene. These limitations resulted in a starkly reduced number[1] of solved problems. The code for the Phaeaco system is not readily available for testing on new BPs.

The work of Depeweg et al. (2018) revisited the earlier approach of Saito and Nakano (1995), refining their logical relations into a proposal for a "visual language." The foundation for their visual language is a grammar allowing them to manually express the correct rule for a subset of 39 BPs (from the original 100). The required features could be computed from off-the-shelf computer

[1] The webpage https://www.foundalis.com/res/solvprog.htm reports 15 problems solved as of 2006.

Figure 13

A Comparison of the Baseline PATHS Model to an Alternative That Cannot Do Physics Simulations



Note. Axes show the average number of steps required to solve a PBP for the baseline (*X*-axis) and no-physics (*Y*-axis) versions of PATHS, with a cutoff at 2,500. The error bars show standard errors of the means for the simulations. The numbers refer to the specific PBP (see Appendix B, for all problems). The brightness of the inner circle shows the successful solution rate for the baseline model, and the outer circle's brightness shows the accuracy for the no-physics model. PATHS = perceiving and testing hypotheses on structures; PBP = physical Bongard problem. See the online article for the color version of this figure.

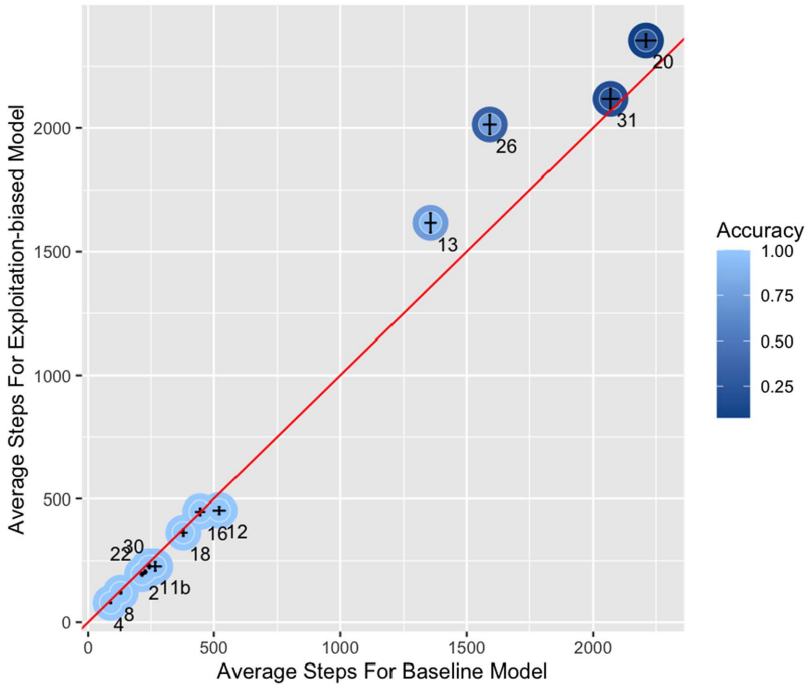
vision algorithms that were more sophisticated than those available to Foundalis more than 10 years earlier. Generating many random instances from the solvable 39 BPs, they were then able to replace the manual construction of the solution rule by an automatic, tree-based classifier that takes the vision-computed features as input. This solver could automatically solve 35 of the 39 problems. However, by the nature of their approach, the remaining 61 were unsolvable a priori as a result of limitations in the expressivity of the constructed visual language.

The recent work by Yun et al. (2020) tries to minimize such limitations of the expressivity of a prechosen symbolic language. To this end, they replace the grammar by a CNN (4 + 1 layers) that transforms each BP scene into a lower dimensional feature vector. This is achieved through a pretraining task requiring the network to learn to classify typical shape constituents (such as line segments, arcs, or n-gons) of a Bongard scene into their shape category (25 classes). By applying the resulting pretrained network to the 12 scenes of a BP it transforms the task of solving the BP into finding a rule for splitting the 12 points (e.g., scenes) in a 25d feature space correctly into the subsets that correspond to the left and right scenes. For finding the rules, they compare two different algorithms: a depth-1 classification tree (equivalent to seeking a single feature that can split each BP's 12 feature points by thresholding) and a linear regression seeking a separating hyperplane in the pretrained feature space—which makes the classifier much more flexible but less interpretable. However, the reported classification rates are rather disappointing (close to 1%). As a remedy, the authors augment the feature space by additional features taken from the hidden layers of the CNN. In this way, they can raise the performance of the regression-based algorithm to close to 100% when they include all features of the last two hidden layers. The first algorithm can improve to about 30% when the hidden features of all layers are included. They also discuss how this impacts the human interpretability of the rules. When they visualize the finally adopted feature as a 2D activity map over the input image, the activity distribution often highlights locations where humans can find interpretable features for building a human-interpretable rule for the inspected BP. Thus, while being very different from a symbolic rule, the created representation can aid a human trying to find such a rule, in the spirit of machines supporting human-in-the-loop problem solving (Zanzotto, 2019).

The authors also consider the generalization (they call it "robustness") of their classifiers by providing for each of the training BPs an additional left and right scenes (hand crafted according to the BP's human-interpretable rule). While both algorithms can improve on the expected 50% generalization ability of a random classifier, they

Figure 14

A Comparison of the Baseline PATHS Model to an Alternative That Is Biased to Apply Already Discovered Descriptions to New Scenes



Note. Axes show the average number of steps required to solve a PBP for the baseline (*X*-axis) and exploitation-biased (*Y*-axis) versions of PATHS. The error bars show standard errors of the means for the simulations. The numbers refer to the specific PBP (see Appendix B, for all problems). The brightness of the inner circle shows the successful solution rate for the baseline model, and the outer circle's brightness shows the accuracy for the exploitation-biased model. PATHS = perceiving and testing hypotheses on structures; PBP = physical Bongard problem. See the online article for the color version of this figure.

do so only slightly (63% and 55% for the first and second algorithms, respectively), indicating a significant overfitting problem for both approaches.

Solving BPs is challenging due to the very small number of examples that exemplify a concept. It is also challenging because of the relative paucity of BPs, compared to, for example, the number of labeled images or large text corpora. At present, the existing domain of BPs consists of only a few hundred readily available for researchers. This, too, poses a problem for AI systems that operate in a primarily data-driven way (Depeweg et al., 2018; Yun et al., 2020) and not generally for humans who reason well from small sets of examples. To relieve this constraint at the domain level and, thereby, make a wider range of machine learning methods applicable, Nie et al. (2020) propose a new visual data set ("BONGARD-LOGO") that is inspired from the original BPs but can be generated in a scalable way from a set of hand-coded programs. Parametric variation within a program allows an unlimited number of instances of each concept. By juxtaposing positive and negative examples of sampled concepts, this data set offers BP-like benchmark problems, but with an arbitrary number of samples from its domain. This allows the authors to evaluate a wider number of machine learning algorithms against human performance. They find that even after relieving the data sparsity at the domain level with their BONGARD-LOGO data set, a large performance gap remains between current machine learning algorithms and human performance.

Compared to the above models, PATHS starts with input images encoded in SVG format, which essentially amounts to assuming a vision system that can perform perfect line and area detection and deliver its results in terms of correspondingly parameterized primitives; a task that can be solved with modern computer vision software for clean images (Depeweg et al., 2018). PATHS is able to work with very few scenes, six or fewer per side, provided that the constructor of a problem is careful to provide a set of scenes that is relatively unambiguous in its classification rule (Shafto et al., 2014). With our focus on dynamics and physics, our tested problems are no longer a subset of Bongard's 100 classical problems. Our newly designed PBPs are mostly out of the scope of other AI systems because their solutions typically require *internal simulation*, combined with the computation of *dynamical features*, for example, the behavior of an object under "imagined" and then simulated physical perturbation.

Although PATHS shares with some of these systems (Depeweg et al., 2018; Saito & Nakano, 1995) the internal use of explicit rules to organize a search through a suitable rule space, most of the PBPs are beyond the expressivity of the above-discussed earlier systems. This applies even for the recent deep CNN approach of Yun et al. (2020;

described above), which is not only negatively impacted by overfitting problems but also restricted by its architecture to the learning of static features. Their transfer learning approach would need to be significantly extended to learn dynamical features. These would require more powerful architectures, such as recurrent or long-short-term-memory networks, trained on more sophisticated data sets. Even if this could be achieved, the goal of producing human-understandable rules would be more difficult to achieve with features generated from recurrent or long-short-term-memory networks than for better understood feedforward CNNs (Yun et al., 2020). However, a recurrent network approach might be able to generate, similar to PATHS, solutions in terms of fast and iterative *internal dynamics*. Although difficult to analyze the resulting internal structures that emerge, such a feat would open up the possibility of a deeper comparison between such a distributed approach and the more symbolic and directly interpretable processing within PATHS (Piantadosi, 2020). Questions could be explored, such as whether a subset of the symbolic operations can be seen as an abstraction of the subsymbolic solution or whether a deep neural network creates an entirely different kind of solution.

One unique contribution of PATHS relative to these other systems is the sophistication of its relation processing. The above-reviewed systems construct logical combinations of features such as *white and circle*, but they lack explicit processes for discovering new relations among separate objects such as *triangle above circle* or *black objects larger than white objects*. Many of the original BPs require spatial or featural relations to be apprehended, and the hypothesis underlying PATHS is that these relations will need to be explicitly represented to achieve robust generalization of a classifier to new instances (Gentner & Asmuth, 2019). This property also sets them apart from the recent data set "BONGARD-LOGO", where the significance of such relations is strongly altered by excluding several basic geometric features, such as size or distance between shapes, from being relevant for a concept (Nie et al., 2020).

A final important difference between PATHS and Deep Learning BP solvers is that even when these latter systems provide an explicit categorization rule, the rule construction phase is strictly after the stage in which features are computed. Instead, PATHS crucially intertwines the process of developing perceptual descriptions with rule construction. An advantage of intertwining perception and concept formation (Austerweil & Griffiths, 2013; Sanborn et al., 2021; Schyns et al., 1998) is that constructing computationally expensive perceptual descriptions is only pursued when supported hypotheses indicate that the effort is likely worthwhile.

Core Requirements for Combining Rule-Based Category Learning With Perceptual Description Construction

The PATHS model is a process-level computational model of human category learning for rule-based categories and can currently solve 13 of the 22 PBPs on which human participants were tested. The PBPs were designed as a challenging problem domain with structured, dynamic physical scenes as the instances that are categorized according to specific rules. The model tightly integrates perception with hypothesis generation and testing: perception drives rule formation and hypothesized rules guide further perception. There is a respectable correlation between PATHS and people on their observed difficulties with different PBPs. Furthermore, PATHS is influenced by similarity and order of presented scene pairs in the same qualitative ways as humans. In this sense, PATHS achieves both its goal of perceiving and learning structured concepts and of capturing an important characteristic of human learning performance.

One possible concern about the PATHS model is that it can currently solve a few PBPs and nothing else. Even if PBPs share important similarities with real-world concept learning tasks, the question of properties and insights from the PATHS model that can be generalized or "exported" into other domains and contexts remains. We go back to the four theoretical commitments that underlie PATHS and discuss why they can be applied to inductive learning more broadly than BPs.

Continual Perception of New Scene Descriptions Over the Course of Category Learning

In many traditional category learning studies, it is easy to imagine that the learner has a full description of the objects to be categorized as soon as they are presented. For example, if eight stimuli are clearly defined by their shape (triangle vs. square), interior line type (solid vs. dotted), and size (large vs. small; Nosofsky et al., 1994), or if colors vary in their saturation and brightness (Nosofsky & Palmeri, 1997), then it is easy and perhaps safe to assume that observers have access to those perceptual features upon stimulus presentation. Other approaches to categorization postulate a gradual process by which perceptual information is accumulated. These models incorporate differential feature salience (Lamberts, 2000) or physiologically plausible neural accumulation processes (Purcell et al., 2010) to provide accounts for the response times required to make categorization decisions. Like these latter models, PATHS incorporates a process by which perceptual descriptions are gradually enriched over time but for different reasons.

For PATHS, one fundamental reason why perceptual descriptions are continually elaborated over time is that building perceptual descriptions is computationally costly. PBPs usefully extend beyond traditional BPs because they underscore the cost of computing descriptions. For example, the determination that a particular scene, if a physics engine were to operate on it, would eventually lead to all of the scenes' objects touching one another involves a costly computation. It requires an internal physics engine to iteratively predict successive frames until an equilibrium or terminal condition is achieved. PATHS's incorporation of a physics engine to infer the properties of a scene is consistent with research suggesting that people can run physics-enabled simulations internally (Allen et al., 2020; Battaglia et al., 2013; T. D. Ullman et al., 2018) and presents a convincing case in which properties such as *support*, *stable*, and *can escape* require something like a physical simulation to determine. While there is clearly a significant computational cost for both humans and PATHS to make such perceptions, the relative costs of perceiving different kind of features are likely different between humans and PATHS—which we do not model explicitly in PATHS. Nevertheless, viewing PBPs through this lens means treating the perception of features in PBP scenes as a costly operation and requires a PBP learning model that minimizes these costs by creating perceptual descriptions only for features that have a reasonable chance of being part of a correct categorization rule. This insight that creating encodings of situations is costly and must be taken into account by a bounded-rational decision maker (Lieder & Griffiths, 2020) applies beyond PBPs. For example,

given the health risks involved in some medical tests, the benefits of collecting additional information about a patient ought to be weighed against the costs of injury from the tests themselves. Likewise, if an investor waits until all of a company's possible financial reports are released, they may miss a rare investment opportunity. The kind of contingent and continual perceptual description construction process at the heart of PATHS is applicable to many situations where acquiring information is costly because of time, labor, or computational resources.

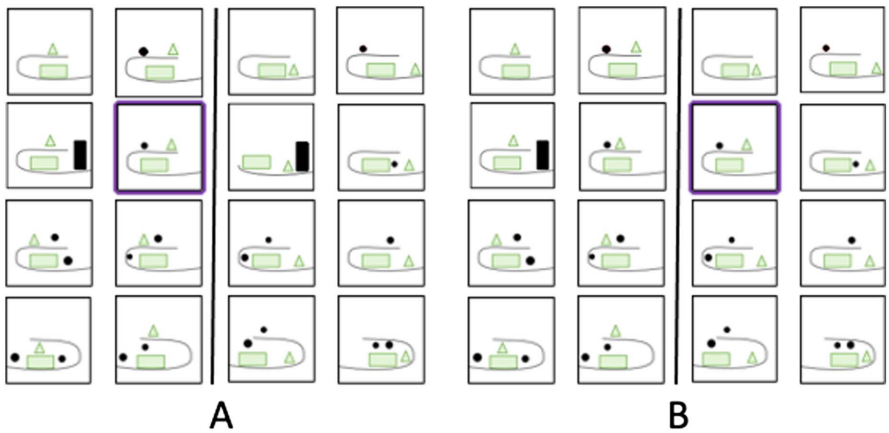
A second reason why perceptual descriptions are continually created as PATHS works to solve a PBP is that there is an open-ended set of possible descriptions that could potentially be built. A typical kind of description formed by PATHS while working on a PBP is ∃ (*objects that are close to (large objects) at the end*), translatable into English as "At the end of the physics simulation, there is at least one object that is close to a large object." Given the composed and complex nature of this description, there are an exceedingly large number of other descriptions that could alternatively have been created. A major source of PATHS's flexibility in finding category rules is that a grammar is used to construct descriptions out of atomically detected features. A natural outcome of PATHS creating new descriptions by recombining previously computed descriptions is that with ongoing processing, increasingly complex descriptions will tend to be created. Creating complex descriptions at the beginning of a run would be inefficient and wasteful and is avoided by having a continual process of creating new descriptions from old.

The Context-Dependent Nature of Constructing Perceptual Descriptions

In PATHS, perceptual descriptions are not only continually formed but they are also contingently formed based on descriptions that have been formed for other scenes. Complex descriptions that would be normally be very unlikely to be constructed can be readily formed if they are first established in another simpler scene. PATHS incorporates three levels of context with increasing scope. At the smallest, *within-scene* level, a feature is more likely to be detected in an object if the feature has been detected in other objects within the scene, or if the object has already been highlighted by having several other features noticed for it. At an intermediate, *within-pair* level, once a feature or relation has been noticed within a scene, it will be checked in the other scene with which it is paired. This is the reason why the immediate temporal and spatial context has a large effect on the difficulty of inducing a categorization rule. At the largest, *within-problem* level, PATHS keeps a running list of all selectors and hypotheses that it has tried. Selectors and hypotheses that have been noticed for many scenes will tend to be noticed for other scenes. This last contextual influence is key to PATHS efficiently finding descriptions that apply to all of the scenes within a category.

There are several ways in which PATHS can give the same scene very different interpretations depending on context. First, groups of objects within a scene can be created not only based upon their spatial proximity but also because they are all picked out by the same selector. For example, a black circle can be placed in the same group with a set of circles based on shape, with a set of black objects based on color, or with white squares if it is sufficiently close to them. Selectors are flexible because they can also be based on relations or conjunctions of features. Second, an object in a scene could be treated as the main object or part of the background. When relations are computed between objects, one object is assigned to be the target object and another is the reference object. Thus, a simple scene could either be described as *left-of*(*triangle, circle*) or *right-of*(*circle, triangle*), and only the former would combine well with another description *left-of*(*white thing, circle*) to form *left-of*(*white triangle, circle*). Third, many features and all relations in PATHS have adaptable thresholds so that, for example, the same situation could be described as "fairly stable" or "fairly unstable." For a more complete example of how this context dependency allows PATHS to solve a PBP, consider the two problems in Figure 15. The PBPs

Figure 15
Context Sensitivity of Perception



Note. The scenes highlighted in purple are the same in Problems A and B, and yet PATHS can form the solution "Left side scenes have a triangle above a rectangle; Right side scenes have a triangle to the right of a rectangle." This is possible because the threshold for detecting *above* and *right* spatial relations depends upon the context provided by the other scenes' descriptions. PATHS = perceiving and testing hypotheses on structures. See the online article for the color version of this figure.

presented in Panels A and B are both solved by PATHS with the rule: "Left side scenes have a triangle above a rectangle; Right side scenes have a triangle to the right of a rectangle." For this to transpire, the identical scene, highlighted in purple, which appears in both problems, must be interpreted as containing a triangle above a rectangle in A but a triangle to the right of a rectangle in B. PATHS is able to give conflicting interpretations to the scene because the thresholds for applying spatial relations such as *above* and *right* are adapted to the context provided by the other scenes. By relaxing the threshold needed to detect a *right-of* in B, PATHS is able to give the highlighted scene the same description that it gives the other scenes in its category. Reminiscent of how Medin et al. (1993) found that the same ambiguous figure is interpreted by people as possessing three prongs when compared to a three-pronged object and as possessing four prongs when compared to a four-pronged object, so PATHS gives incompatible interpretations of a scene to make it align better with other scenes in its comparison group.

Hypothesis Construction by Combining Descriptions Into Compound Rules

PATHS is not unique in possessing a mechanism for recursively creating compound rules out of simpler expressions. Many cognitive models have the ability to express new concepts by applying built-in operations to a finite set of primitive components in order to generate an open-ended set of descriptions (Goodman et al., 2008, 2015; Kemp, 2012; Piantadosi et al., 2016). The power of these compositional systems is that grammars for composing symbols can express higher order ideas that would be very difficult to represent directly in the senses, and they permit a combinatorial explosion of possible ideas (Piantadosi, 2020; Rule et al., 2020). In linguistics, combinatorial systems have been proposed to represent open-class verbs like "kill" in terms of the components *cause* and *die*. Within PATHS, physical relational concepts provide good examples of the generative power of combinatorics. For example, PATHS can coarsely capture the notion of "circle is free to escape" (see PBP 31) by building up the hypothesis that "there are can-move-up circles in all left scenes." The semantics of the "can-move-up" component are, in turn, grounded by the execution of a perceptual simulation that tests whether the object can easily move out of the scene when at least one directional force is applied to it. The physics simulation grounds the notion of stability in terms of whether small perturbations to the initial scene result in large end-state differences. Likewise, the notion of *X* supporting *Y* is grounded in terms of the conjunction of *X* being below *Y* and *Y* touching *X*. This physical grounding of *support* is clearly not general enough to explain other uses of "support" such as data supporting an argument or a parent supporting a child (Regier, 1996). However, PATHS at least provides a working computational model capable of representing some relational concepts that would normally be considered high level and abstract.

A legitimate question remains: "Why build a model of rule-based category learning if most human categories are not structured by rules?" According to prototype approaches, concepts are organized around family resemblances rather than features that are individually necessary and jointly sufficient for categorization (Lakoff, 1987; Rosch & Mervis, 1975). Prototype accounts can naturally accommodate observations that are awkward for rule-based approaches, such as people's general difficulty describing rules that make up natural concepts such as *robin*, *chair*, and *mother* (Goldstone et al., 2018). However, category learning systems that devise rules are still, arguably even increasingly, important for several reasons. First, empirical results on category learning often produce findings that are more compellingly explained by people forming rules rather than learning prototypes or storing instances (Nosofsky & Palmeri, 1998; Piantadosi & Jacobs, 2016). Second, for AI systems, a practically and theoretically important goal is developing intelligent systems that can explain or justify the decisions that they make (Samek & Müller, 2019). For example, if a doctor cannot understand why a deep learning system categorizes a patient's growth as malignant, then the doctor is less likely to trust the categorization and may not implement the best course of treatment. Third, teachers who strive to efficiently impart knowledge to students plainly benefit if they are able to give their students rules, even if they are imperfect. The student of German who learns from their teacher that nouns that end in "e" usually have a feminine grammatical gender may make some mistakes but will very often correctly guess the gender of a word never before seen. Fourth, rule-based reasoning allows for stronger inferencing than is possible with systems that use overall similarity (Sloman, 1996). For example, Cuba may be similar to Jamaica, because of their climates, and similar to Russia politically, but these similarities do not sanction the inference that Jamaica is similar to Russia (Tversky, 1977). If a search engine is going to produce the correct answer to a query such as "How many U.S. cities that have a population over 500,000 are located North of Latitude 40?" then it is likely going to need to have the capacity to represent rules as conjunctions of conditions.

Bidirectional Interactions Between Perceiving New Aspects of Scenes and Constructing Rules That Distinguish Categories

An ongoing key issue for cognition is how people's low-level perceptual systems are connected to their high-level conceptual and reasoning systems. In traditional symbolic AI systems, this connection is not sought, and computation only involves symbol-to-symbol transformations. However, others have argued that if symbolic conceptual and reasoning processes are not interfaced with perception and action, then these high-level systems can neither get information from nor affect the world (Harnad, 1990). Proponents of embodied cognition emphasize that abstract conceptual representations gain their flexibility and richness from their foundation in perceptual systems (Barsalou, 1999, 2008; Goldstone & Barsalou, 1998). Likewise, researchers in robotics have had to grapple with the need to have agents that interact with their environments on a sensorimotor level and yet need concepts that allow them to transfer what they have learned to different environments (Lázaro-Gredilla et al., 2019).

BPs offer an ideal domain for studying the interface between perception and concepts because they require both the flexible perception of features from complex visuospatial inputs and conceptual apparatus to organize descriptions into explicit rules (Edelman & Shahbazi, 2012; Hofstadter, 1979; M. Mitchell, 2019; Piantadosi, 2020). While PATHS hardly solves the general problem of how to interface low-level perceptual inputs to conceptual symbols, it does offer a working computational model that solves nontrivial problems requiring both perception and symbolic representations.

Critical to PATHS's operation are bidirectional interactions between perception and hypothesis formation. When features, groups, and relations are identified in a scene in the form of selectors, they spur the formation of hypotheses. Hypotheses are candidate rules for distinguishing between the scenes on the two sides of a BP. These hypotheses, in turn, direct PATHS to search for additional confirmatory and disconfirmatory evidence. From the vast number of costly descriptions that could potentially be built for a scene, a much smaller set of descriptions is thereby prioritized for construction. Efficient learning from complex examples relies on good choices about the order in which features and rules are explored. Modeling the iterative and concurrent nature of perception and rule construction allows these processes to mutually guide these choices.

We take these four core requirements to be domain-general principles for category learning in situations where new perceptual descriptions must be created to support the categorization. It might be argued that much of PATHS' processing is task specific, useful for PBPs but not generalizing well to other tasks. In response, we would argue that many of the specific interpretative processes developed for PBPs have applications for other problems. Some of these generally applicable mechanisms include the graded determination of spatial relations, the prioritized testing of descriptions developed for one scene on other scenes, the creation of new descriptions by composing existing descriptions and grouping objects in a scene into clusters based on expressions using quantifiers. Beyond PBPs, these mechanisms have direct relevance to problems in the abstraction and reasoning corpus (Chollet, 2019), visual analogical reasoning (M. Mitchell, 2021), and Raven's progressive matrices problems (Hersche et al., 2023). Some of PATHS' description generation mechanisms are restricted in their relevance to visual perception and simulation rather than being completely abstract and amodal, but concepts that involve spatialized perception and simulation are rife throughout human experience (Allen et al., 2020; Battaglia et al., 2013; T. D. Ullman et al., 2017). Moreover, nontrivial cases of creating new scene descriptions may necessarily involve perceptually constrained interpretative processes such as the ones found in PATHS (S. Ullman, 1987). Eventually, these processes may themselves be learnable (Lázaro-Gredilla et al., 2019).

Conclusion

BPs provide an elegant context for exploring the complex process of finding rules that organize rich perceptual inputs into categories. Solving these problems requires a cognitive system to bridge the semantic gap between low-level perception and high-level conceptualization (M. Mitchell, 2020). The PBPs that we introduce augment the set of typical BPs so as to emphasize the need for solvers to engage in costly perceptual simulations that approximately follow physical laws. PBPs are attractive due to their open feature space, their dynamic and structured content, and the fact that they allow for easy manipulation of the similarity of scenes presented next to each other. While studies with human participants serve as a comparison for the model, they also directly advanced our understanding of how the mode of presentation and similarity of instances influence human concept learning. These studies show the benefit of cross-category comparison of similar relative to dissimilar PBP scenes and the benefit of within-category comparison of dissimilar relative to similar PBP scenes. The components of PATHS that perceive dynamic physical attributes of scenes such as stability, support, and movability by using a physics engine to perform counterfactual reasoning can be reused in other work that explores the perceptual grounding of high-level concepts. PATHS also presents a system for integrating perceptual processes tightly with higher level cognitive processes. This integration is required across fields as diverse as active learning, optimal experiment design, analogy making, and memory retrieval. Within the field of concept learning itself, there has been a tendency to treat the perceptual encoding process as separate from, and completed before, the process of constructing characterizations of concepts. PBPs show the conceptual problems that arise when these processes are kept separate, and PATHS provides a model of how they can be effectively integrated.

References

Aha, D. W., & Goldstone, R. L. (1992). *Concept learning and flexible weighting* [Conference session]. Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, Hillsdale, New Jersey, United States.

Allen, K. R., Smith, K. A., & Tenenbaum, J. B. (2020). Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences*, *117*(47), 29302–29310. https://doi.org/10.1073/pnas.1912341117

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, *98*(3), 409–429. https://doi.org/10.1037/0033-295X.98.3.409

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.

Arciszewski, T., Michalski, R. S., & Wnek, J. (1995). *Constructive induction: The key to design creativity* [Conference session]. Preprints of the Third International Round-Table Conference on Computational Models of Creative Design, Heron Island, Queensland, Australia.

Austerweil, J. L., & Griffiths, T. L. (2013). A nonparametric Bayesian framework for constructing flexible feature representations. *Psychological Review*, *120*, 817–851. https://doi.org/10.1037/a0034194

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*(4), 577–660. https://doi.org/10.1017/s0140525x99002149

Barsalou, L. W. (2008). Grounded cognition. *Annual Review Psychology*, *59*, 617–645. https://doi.org/10.1146/annurev.psych.59.103006.093639

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, *110*(45), 18327–18332. https://doi.org/10.1073/pnas.1306572110

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Bloch, I. (1999). Fuzzy relative position between objects in image processing: A morphological approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(7), 657–664. https://doi.org/10.1109/34.777378

Bloch, I. (2010). Bipolar fuzzy spatial information: Geometry, morphology, spatial reasoning. In R. Jeansoulin, O. Papini, H. Prade, & S. Schockaert (Eds.), *Methods for handling imperfect spatial information* (pp. 75–102). Springer.

Bongard, M. M. (1970). *Pattern recognition*. Hayden Book Corporation, Spartan Books.

Bruner, J., Goodnow, J., & Austin, G. (1977). *A study of thinking*. R. E. Krieger Publishing Company. https://books.google.com/books?id=Wb1fPwAACAAJ

Brunmair, M., & Richter, T. (2019). Similarity matters: A meta-analysis of interleaved learning and its moderators. *Psychological Bulletin*, *145*(11), 1029–1052. https://doi.org/10.1037/bul0000209

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., & Zhang, Y. (2023). *Sparks of artificial general intelligence: Early experiments with gpt-4*. arXiv.

Carvalho, P. F., & Goldstone, R. L. (2014). Putting category learning in order: Category structure and temporal arrangement affect the benefit of interleaved over blocked study. *Memory & Cognition*, *42*(3), 481–495. https://doi.org/10.3758/s13421-013-0371-0

Carvalho, P. F., & Goldstone, R. L. (2015). The benefits of interleaved and blocked study: Different tasks benefit from different schedules of study. *Psychonomic Bulletin & Review*, *22*(1), 281–288. https://doi.org/10.3758/s13423-014-0676-4

Carvalho, P. F., & Goldstone, R. L. (2022). A computational model of context-dependent encodings during category learning. *Cognitive Science*, *46*(4), Article e13128. https://doi.org/10.1111/cogs.13128

Chang, N., Koedinger, K. R., & Lovett, M. C. (2003). *Learning spurious correlations instead of deeper relations* [Conference session]. Proceedings of the 25th Cognitive Science Society, Boston, Massachusetts, United States.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29* (pp. 2172–2180). Curran Associates.

Chollet, F. (2019). *On the measure of intelligence*. arXiv. http://arxiv.org/abs/1911.01547

Christie, S., & Gentner, D. (2010). Where hypotheses come from: Learning new relations by structural alignment. *Journal of Cognition and Development*, *11*(3), 356–373. https://doi.org/10.1080/15248371003700015

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181–204. https://doi.org/10.1017/S0140525X12000477

Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*(1481), 933–942. https://doi.org/10.1098/rstb.2007.2098

Dehaene, S., Izard, V., Pica, P., & Spelke, E. (2006). Core knowledge of geometry in an amazonian indigene group. *Science*, *311*(5759), 381–384. https://doi.org/10.1126/science.1121739

Depeweg, S., Rothkopf, C. A., & Jäkel, L. (2018). *Solving bongard problems with a visual language and pragmatic reasoning*. arXiv.

Dietterich, T. G., Domingos, P., Getoor, L., Muggleton, S., & Tadepalli, P. (2008). Structured machine learning: The next ten years. *Machine Learning*, *73*(1), 3–23. https://doi.org/10.1007/s10994-008-5079-1

Diettrich, T. G., & Michalski, R. S. (1985). Discovering patterns in sequences of events. *Artificial Intelligence*, *25*(2), 187–232. https://doi.org/10.1016/0004-3702(85)90003-7

Doug Rohrer, R. F. D., & Hartwig, M. K. (2020). The scarcity of interleaved practice in mathematics textbooks. *Educational Psychology Review*, *32*(3), 873–883. https://doi.org/10.1007/s10648-020-09516-2

Edelman, S., & Shahbazi, R. (2012). Renewing the respect for similarity. *Frontiers in Computational Neuroscience*, *6*, Article 45. https://doi.org/10.3389/fncom.2012.00045

Esfandiari, N., Babavalian, M. R., Moghadam, A.-M. E., & Tabar, V. K. (2014). Knowledge discovery in medicine: Current issue and future trend. *Expert Systems With Applications*, *41*(9), 4434–4463. https://doi.org/10.1016/j.eswa.2014.01.011

Esteva, A., Kuprel, B., & Novoa, R. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118. https://doi.org/10.1038/nature21056

Flennerhag, S., Schroecker, Y., Zahavy, T., van Hasselt, H., Silver, D., & Singh, S. (2022). *Bootstrapped meta-learning*. arXiv. https://doi.org/10.48550/arXiv.2109.04504

Forbus, K. D. (1994). Qualitative process theory: Twelve years after. *Artificial Intelligence in Perspective*, *59*(1), 115–123. https://doi.org/10.1016/0004-3702(93)90177-D

Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending sme to handle large-scale cognitive modeling. *Cognitive Science*, *41*(5), 1152–1201. https://doi.org/10.1111/cogs.12377

Foster, N. L., Mueller, M. L., Was, C., Rawson, K. A., & Dunlosky, J. (2019). Why does interleaving improve math learning? The contributions of discriminative contrast and distributed practice. *Memory & Cognition*, *47*(6), 1088–1101. https://doi.org/10.3758/s13421-019-00918-4

Foundalis, H. E. (2006). *Phaeaco: A cognitive architecture inspired by bongard's problems* [Unpublished doctoral dissertation]. Indiana University.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, *13*(1), 3–54. https://doi.org/10.1023/A:1006524209794

Gauthier, I., Tarr, M. J., & Bubb, D. E. (2010). *Perceptual expertise: Bridging brain and behavior*. Oxford University Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170. https://doi.org/10.1207/s15516709cog0702_3

Gentner, D., & Asmuth, J. (2019). Metaphoric extension, relational categories, and abstraction. *Language, Cognition and Neuroscience*, *34*(10), 1298–1307. https://doi.org/10.1080/23273798.2017.1410560

Getoor, L., & Taskar, B. (2007). *Introduction to statistical relational learning*. MIT Press.

Goldstone, R. L. (1996). Isolated and interrelated concepts. *Memory & Cognition*, *24*(5), 608–628. https://doi.org/10.3758/BF03201087

Goldstone, R. L., & Barsalou, L. (1998). Reuniting perception and conception. *Cognition*, *65*(2–3), 231–262. https://doi.org/10.1016/S0010-0277(97)00047-4

Goldstone, R. L., Day, S., & Son, J. Y. (2010). Comparison. In V. G. B. Glatzeder & A. von Müller (Eds.), *Towards a theory of thinking* (Vol. II, pp. 103–122). Springer Verlag GmbH.

Goldstone, R. L., Kersten, A., & Carvalho, P. F. (2018). Categorization and concepts. In J. Wixted (Ed.), *Stevens' handbook of experimental psychology and cognitive neuroscience* (4th ed., Vol. 3, pp. 275–317). Wiley.

Gómez, R. L. (2002). Variability and detection of invariant structure. *Psychological Science*, *13*(5), 431–436. https://doi.org/10.1111/1467-9280.00476

Goodman, N. D., Tenenbaum, J., Feldman, J., & Griffiths, T. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–54. https://doi.org/10.1080/03640210701802071

Goodman, N. D., Tenenbaum, J. B., & Gerstenberg, T. (2015). Concepts in a probabilistic language of thought. In E. Margolis & S. Laurence (Eds.), *The conceptual mind: New directions in the study of concepts* (pp. 623–653). MIT Press.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Halpern, D., Hansen, C., & Riefer, D. (1990). Analogies as an aid to understanding and memory. *Journal of Educational Psychology*, *82*(2), 298–305. https://doi.org/10.1037/0022-0663.82.2.298

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*(1–3), 335–346. https://doi.org/10.1016/0167-2789(90)90087-6

Hersche, M., Zeqiri, M., Benini, L., Sebastian, A., & Rahimi, A. (2023). A neuro-vector-symbolic architecture for solving Raven's progressive matrices. *Nature Machine Intelligence*, *5*, 363–375. https://doi.org/10.1038/s42256-023-00630-8

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). *Beta-vae: Learning basic visual*

concepts with a constrained variational framework [Conference session]. International Conference on Learning Representations, Toulon, France.

Hofstadter, D. R. (1979). *Gödel, Escher, Bach: An eternal golden braid*. Harvester Press.

Hubbard, T. L. (2005). Representational momentum and related displacements in spatial memory: A review of the findings. *Psychonomic Bulletin & Review*, *12*(5), 822–851. https://doi.org/10.3758/BF03196775

Hudelot, C., Atif, J., & Bloch, I. (2008). Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, *159*(15), 1929–1951. https://doi.org/10.1016/j.fss.2008.02.011

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*(3), 427–466. https://doi.org/10.1037/0033-295X.104.3.427

Hunt, E. B., Marin, J., & Stone, P. J. (1966). *Experiments in induction*. Academic Press.

Johnson, S. (2006). *The ghost map: The story of London's most terrifying epidemic—And how it changed science, cities, and the modern world*. Riverhead Books.

Kang, S. H., & Pashler, H. (2012). Learning painting styles: Spacing is advantageous when it promotes discriminative contrast. *Applied Cognitive Psychology*, *26*(1), 97–103. https://doi.org/10.1002/acp.1801

Kemp, C. (2012). Exploring the conceptual universe. *Psychological Review*, *119*(4), 685–722. https://doi.org/10.1037/a0029347

Kornell, N., & Bjork, R. A. (2008). Learning concepts and categories: Is spacing the "enemy of induction"? *Psychological Science*, *19*(6), 585–592. https://doi.org/10.1111/j.1467-9280.2008.02127.x

Kruschke, J. K. (1992). Alcove: An exemplar-based connectionist model of category learning. *Psychological Review*, *99*(1), 22–44. https://doi.org/10.1037/0033-295X.99.1.22

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338. https://doi.org/10.1126/science.aab3050

Lake, B. M., Ullman, T. D., Tenenbaum, J. T., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*, Article E1. https://doi.org/10.1017/S0140525X1500062X

Lakoff, G. (1987). *Women, fire and dangerous things: What categories reveal about the mind*. University of Chicago Press.

Lamberts, K. (2000). Information-accumulation theory of speeded categorization. *Psychological Review*, *107*(2), 227–260. https://doi.org/10.1037/0033-295X.107.2.227

Lara-Dammer, F., Hofstadter, D. R., & Goldstone, R. L. (2019). A computational model of scientific discovery in a very simple world aiming at psychological realism. *Journal of Experimental & Theoretical Artificial Intelligence*, *31*(4), 637–658. https://doi.org/10.1080/0952813X.2019.1592234

Lázaro-Gredilla, M., Lin, D., Guntupalli, J. S., & George, D. (2019). Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs. *Science Robotics*, *4*(26), Article eaav315. https://doi.org/10.1126/scirobotics.aav3150

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, *43*, Article E1. https://doi.org/10.1017/S0140525X1900061X

Loewenstein, J., & Gentner, D. (2005). Relational language and the development of relational mapping. *Cognitive Psychology*, *50*(4), 315–353. https://doi.org/10.1016/j.cogpsych.2004.09.004

Love, B. C., Medin, D. L., & Gureckis, T. M. (2004). Sustain: A network model of category learning. *Psychological Review*, *111*(2), 309–332. https://doi.org/10.1037/0033-295X.111.2.309

Ludwin-Peery, E., Bramley, N. R., Davis, E., & Gureckis, T. M. (2020). Broken physics: A conjunction-fallacy effect in intuitive physical reasoning. *Psychological Science*, *31*(12), 1602–1611. https://doi.org/10.1177/0956797620957610

Markman, A. B., & Gentner, D. (1993). Structural alignment during similarity comparisons. *Cognitive Psychology*, *25*(4), 431–467. https://doi.org/10.1006/cogp.1993.1011

McBratney, A., Mendonca Santos, M., & Minasny, B. (2003). On digital soil mapping. *Geoderma*, *117*(1), 3–52. https://doi.org/10.1016/S0016-7061(03)00223-4

McCloskey, M., & Kohl, D. (1983). Naive physics: The curvilinear impetus principle and its role in interactions with moving objects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *88*(1), 146–156. https://doi.org/10.1037/h0030693

Medin, D. L., Goldstone, R., & Gentner, D. (1993). Respects for similarity. *Psychological Review*, *100*(2), 254–278. https://doi.org/10.1037/0033-295X.100.2.254

Medin, D. L., & Ross, B. (1989). The specific character of abstract thought: Categorization, problem solving, and induction. In R. J. Sternberg (Ed.), *Advances in the psychology of human intel ligence* (Vol. 5, pp. 189–223). Lawrence Erlbaum.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*(3), 207–238. https://doi.org/10.1037/0033-295X.85.3.207

Medin, D. L., Wattenmaker, W. D., & Michalski, R. S. (1987). Constraints and preferences in inductive learning: An experimental study of human and machine performance. *Cognitive Science*, *11*(3), 299–339. https://doi.org/10.1207/s15516709cog1103_3

Michalski, R., & Chilausky, R. (1980). Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology. *International Journal of Man-Machine Studies*, *12*(1), 63–87. https://doi.org/10.1016/S0020-7373(80)80054-X

Mitchell, M. (1993). *Analogy-making as perception*. MIT Press/Bradford Books.

Mitchell, M. (2019). *Artificial intelligence: A guide for thinking humans*. Farrar, Straus and Giroux.

Mitchell, M. (2020). On crashing the barrier of meaning in artificial intelligence. *AI Magazine*, *41*(2), 86–92. https://doi.org/10.1609/aimag.v41i2.5259

Mitchell, M. (2021). Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, *1505*(1), 79–101. https://doi.org/10.1111/nyas.14619

Mitchell, M., & Krakauer, D. C. (2023). The debate over understanding in AI's large language models. *Proceedings of the National Academy of Sciences*, *120*(13), Article e2215907120. https://doi.org/10.1073/pnas.2215907120

Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers University.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, *18*(2), 203–226. https://doi.org/10.1016/0004-3702(82)90040-6

Muggleton, S. (1992). *Inductive logic programming*. Springer.

Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, *19*(Suppl. 1), 629–679. https://doi.org/10.1016/0743-1066(94)90035-3

Mumford, M. D., & McIntosh, T. (2017). Creative thinking processes: The past and the future. *The Journal of Creative Behavior*, *51*(4), 317–322. https://doi.org/10.1002/jocb.197

Murphy, G. L., & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review*, *92*(3), 289–316. https://doi.org/10.1037/0033-295X.92.3.289

Nie, W., Yu, Z., Mao, L., Patel, A. B., Zhu, Y., & Anandkumar, A. (2020). BONGARD-LOGO: A new benchmark for human-level concept learning and reasoning. *Advances in Neural Information Processing Systems*, *33*, 16468–16480. https://research.nvidia.com/sites/default/files/publications/Paper.pdf

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*(1), 104–114. https://doi.org/10.1037/0278-7393.10.1.104

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, *115*(1), 39–57. https://doi.org/10.1037/0096-3445.115.1.39

Nosofsky, R. M., Gluck, M. A., Palmeri, T. J., & McKinley, S. C. (1994). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory & Cognition*, *22*(3), 352–369. https://doi.org/10.3758/BF03200862

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review*, *104*(2), 266–300. https://doi.org/10.1037/0033-295X.104.2.266

Nosofsky, R. M., & Palmeri, T. J. (1998). A rule-plus-exception model for classifying objects in continuous-dimension spaces. *Psychonomic Bulletin & Review*, *5*(3), 345–369. https://doi.org/10.3758/BF03208813

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, *23*(2), 324–354. https://doi.org/10.1037/0278-7393.23.2.324

Piantadosi, S. T. (2020). The computational origin of representation. *Minds and Machines*, *31*, 1–58. https://doi.org/10.1007/s11023-020-09540-9

Piantadosi, S. T., & Jacobs, R. (2016). Four problems solved by the probabilistic language of thought. *Current Directions in Psychological Science*, *25*, 54–59. https://doi.org/10.1177/0963721415609581

Piantadosi, S. T., Tenenbaum, J., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, *123*(4), 392–424. https://doi.org/10.1037/a0039980

Purcell, B. A., Heitz, R. P., Cohen, J. Y., Schall, J. D., Logan, G. D., & Palmeri, T. J. (2010). Neurally constrained modeling of perceptual decision making. *Psychological Review*, *117*(4), 1113–1143. https://doi.org/10.1037/a0020311

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/BF00116251

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). *Hierarchical text-conditional image generation with clip latents*. arXiv.

Regier, T. (1996). *The human semantic potential: Spatial language and constrained connectionism*. MIT Press.

Regier, T., & Carlson, L. A. (2001). Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, *130*(2), 273–298. https://doi.org/10.1037/0096-3445.130.2.273

Rittle-Johnson, B., Star, J., & Durkin, K. (2020). How can cognitive-science research help improve education? The case of comparing multiple strategies to improve mathematics learning and teaching. *Current Directions in Psychological Science*, *29*(6), 599–609. https://doi.org/10.1177/0963721420969365

Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*(4), 573–605. https://doi.org/10.1016/0010-0285(75)90024-9

Ross, D. A., Tamber-Rosenau, B. J., Palmeri, T. J., Zhang, J., Xu, Y., & Gauthier, I. (2018). High resolution fMRI reveals configural processing of cars in right anterior fusiform face area of car experts. *Journal of Cognitive Neuroscience*, *30*(7), 973–984. https://doi.org/10.1162/jocn_a_01256

Rost, G. C., & McMurray, B. (2009). Speaker variability augments phonological processing in early word learning. *Developmental Science*, *12*(2), 339–349. https://doi.org/10.1111/j.1467-7687.2008.00786.x

Roy, B. C., Frank, M. C., DeCamp, P., Miller, M., & Roy, D. (2015). Predicting the birth of a spoken word. *Proceedings of the National Academy of Sciences*, *112*(41), 12663–12668. https://doi.org/10.1073/pnas.1419773112

Rule, J. S., Tenenbaum, J. B., & Piantadosi, S. T. (2020). The child as hacker. *Trends in Cognitive Science*, *24*(11), 900–915. https://doi.org/10.1016/j.tics.2020.07.005

Saito, K., & Nakano, R. (1995). Adaptive concept learning algorithm: Rf4. *Transactions of IPSJ*, *36*(4), 832–839.

Samek, W., & Müller, K.-R. (2019). Towards explainable artificial intelligence. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, & K.-R. Müller (Eds.), *Explainable AI: Interpreting, explaining and visualizing deep learning* (pp. 5–22). Springer International Publishing.

Sanborn, A. N., Heller, K., Austerweil, J. L., & Chater, N. (2021). REFRESH: A new approach to modeling dimensional biases in perceptual similarity and categorization. *Psychological Review*, *128*(6), 1145–1186. https://doi.org/10.1037/rev0000310

Schyns, P. G., Goldstone, R. L., & Thibaut, J.-P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences*, *21*(1), 1–17. https://doi.org/10.1017/S0140525X98000107

Shafto, P., Goodman, N. D., & Griffiths, T. L. (2014). A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive Psychology*, *71*, 55–89. https://doi.org/10.1016/j.cogpsych.2013.12.004

Silver, D., Schrittwieser, J., & Simonyan, K. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*, 354–359. https://doi.org/10.1038/nature24270

Simon, H. A., & Feigenbaum, E. A. (1964). An information-processing theory of some effects of similarity, familiarization, and meaningfulness in verbal learning. *Journal of Verbal Learning and Verbal Behavior*, *3*(5), 385–396. https://doi.org/10.1016/S0022-5371(64)80007-4

Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin*, *119*, 3–22. https://doi.org/10.1037/0033-2909.119.1.3

Stanley, K. O., & Lehman, J. (2015). *Why greatness cannot be planned: The myth of the objective*. Springer.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018, June). *Learning to compare: Relation network for few-shot learning* [Conference session]. The IEEE conference on computer vision and pattern recognition (cvpr), Salt Lake City, UT, United States.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*(6022), 1279–1285. https://doi.org/10.1126/science.1192788

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352. https://doi.org/10.1037/0033-295X.84.4.327

Twomey, K. E., & Westermann, G. (2018). Curiosity-based learning in infants: A neurocomputational approach. *Developmental Science*, *21*(4), Article e12629. https://doi.org/10.1111/desc.12629

Ullman, S. (1987). Visual routines. In M. A. Fischler & O. Firschein (Eds.), *Readings in computer vision* (pp. 298–328). Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-051581-6.50035-0

Ullman, T. D., Spelke, E., Battaglia, P., & Tenenbaum, J. B. (2017). Mind games: Game engines as an architecture for intuitive physics. *Trends in Cognitive Sciences*, *21*(9), 649–665. https://doi.org/10.1016/j.tics.2017.05.012

Ullman, T. D., Stuhlmüller, A., Goodman, N. D., & Tenenbaum, J. B. (2018). Learning physical parameters from dynamic scenes. *Cognitive Psychology*, *104*, 57–82. https://doi.org/10.1016/j.cogpsych.2017.05.006

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 29, pp. 3630–3638). Curran Associates.

Webb, T. W., Sinha, I., & Cohen, J. D. (2021). *Emergent symbols through binding in external memory*. arXiv. https://doi.org/10.48550/arXiv.2012.14601

Weitnauer, E. (2016). *Interactions between perception and rule-construction in human and machine concept learning* [Doctoral Dissertation]. Universität Bielefeld.

Weitnauer, E., Carvalho, P. F., Goldstone, R. L., & Ritter, H. (2013). *Grouping by similarity helps concept learning* [Conference session].

Proceedings of the Thirty-Fifth Annual Conference of the Cognitive Science Society.

Weitnauer, E., Carvalho, P. F., Goldstone, R. L., & Ritter, H. (2014). *Similarity-based ordering of instances for efficient concept learning* [Conference session]. Proceedings of the Thirty-Sixth Annual Conference of the Cognitive Science Society.

Winston, P. H. (1970). *Learning structural descriptions from examples* (technical report). Massachusetts Institute of Technology.

Wnek, J., & Michalski, R. S. (1994). Hypothesis-driven constructive induction in aq17-hci: A method and experiments. *Machine Learning*, *14*(2), 139–168. https://doi.org/10.1023/A:1022622132310

Yan, V. X., & Sana, F. (2021). The robustness of the interleaving benefit. *Journal of Applied Research in Memory and Cognition*, *10*(4), 589–602. https://doi.org/10.1037/h0101863

Yun, X., Bohn, T., & Ling, C. (2020). A deeper look at bongard problems. In C. Goutte & X. Zhu (Eds.), *Advances in artificial intelligence* (pp. 528–539). Springer International Publishing.

Zanzotto, F. M. (2019). Viewpoint: Human-in-the-loop Artificial Intelligence. *Journal of Artificial Intelligence Research*, *64*, 243–252. https://doi.org/10.1613/jair.1.11345

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., & Song, Y. (2018). Metagan: An adversarial approach to few-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31* (pp. 2365–2374). Curran Associates.

Zulkiply, N., & Burt, J. S. (2013). The exemplar interleaving effect in inductive learning: Moderation by the difficulty of category discriminations. *Memory & Cognition*, *41*(1), 16–27. https://doi.org/10.3758/s13421-012-0238-9

Appendix A

Solutions to Bongard Problems Presented in Main Text

Figure 1 problems:

- Upper left—left side: small objects included; right side: no small objects.
- Upper right—left side: black object is triangle; right side: black object is circle.
- Lower left—left side: line endpoints have same orientation; right side: line endpoints are oriented 90° relative to one another.
- Lower right—left side: the line and point form an isosceles triangle; right side: the line and point form a scalene triangle.

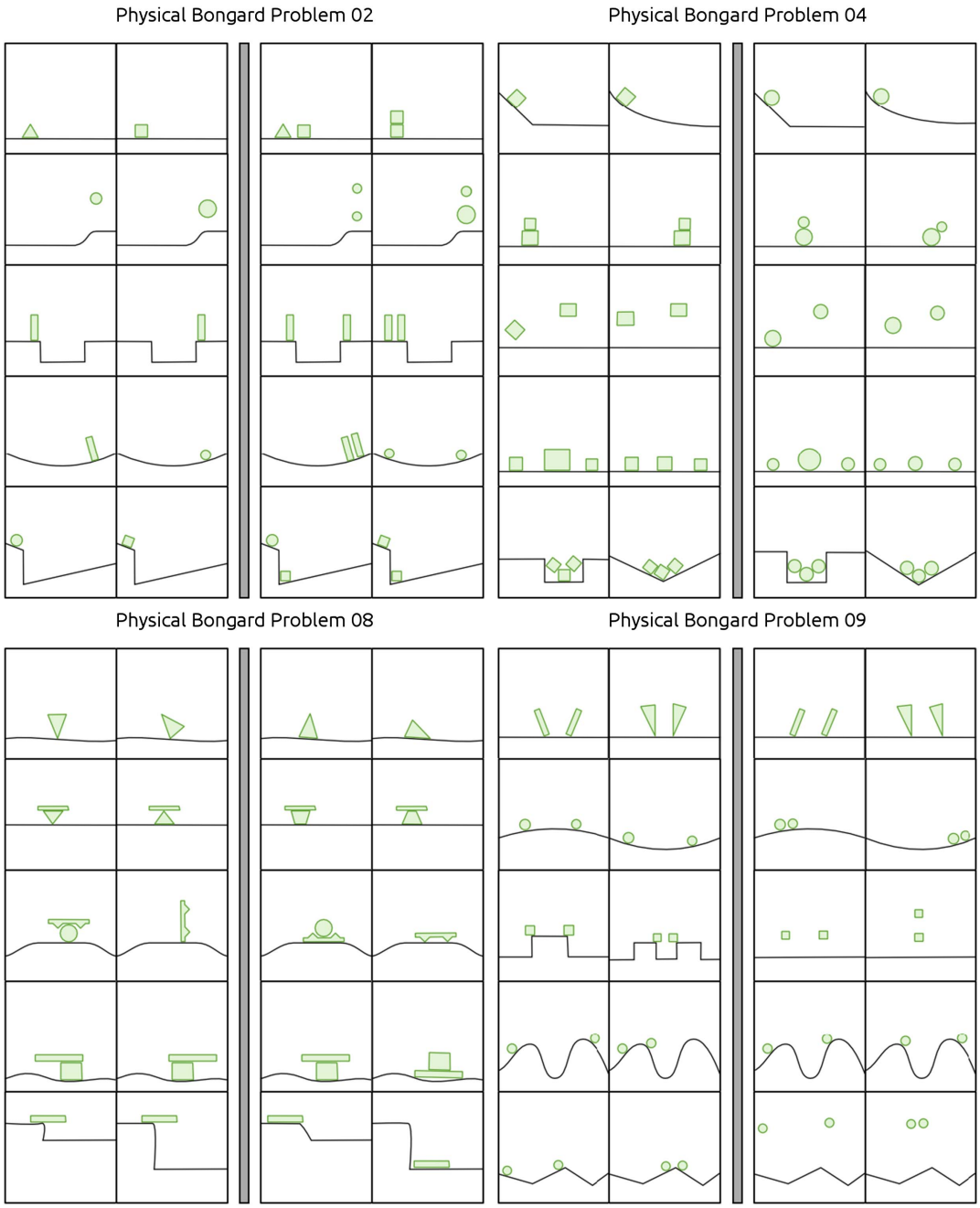
Figure 2 problems:

- Upper left—left side: objects end up touching each other; right side: objects end up apart.
- Upper right—left side: circle does not land between two identical objects; right side: circle lands between two identical objects.
- Lower left—left side: object collide; right side: objects do not collide.
- Lower right—left side: structure is not destroyed; right side: structure is destroyed.

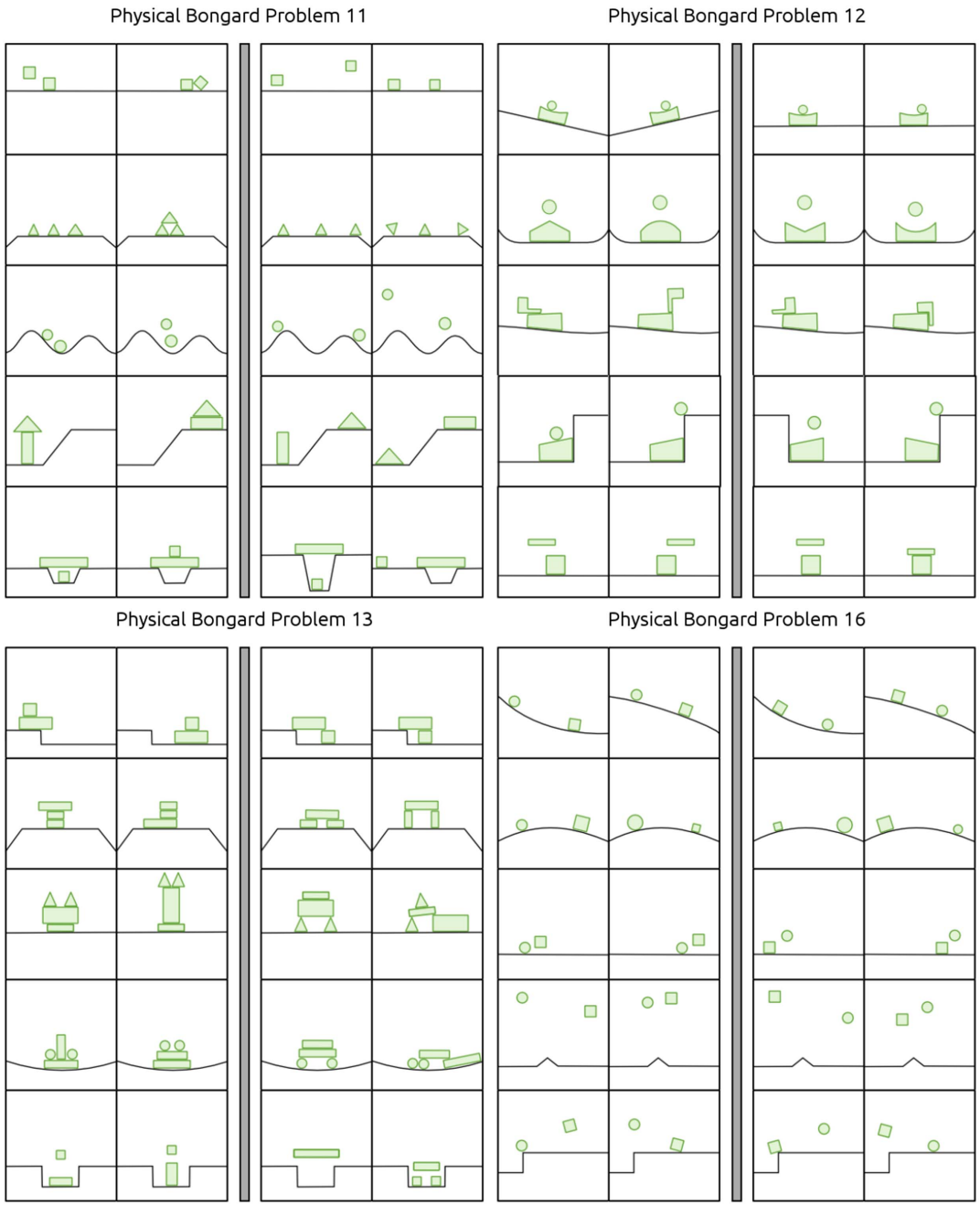
(*Appendices continue*)

Appendix B

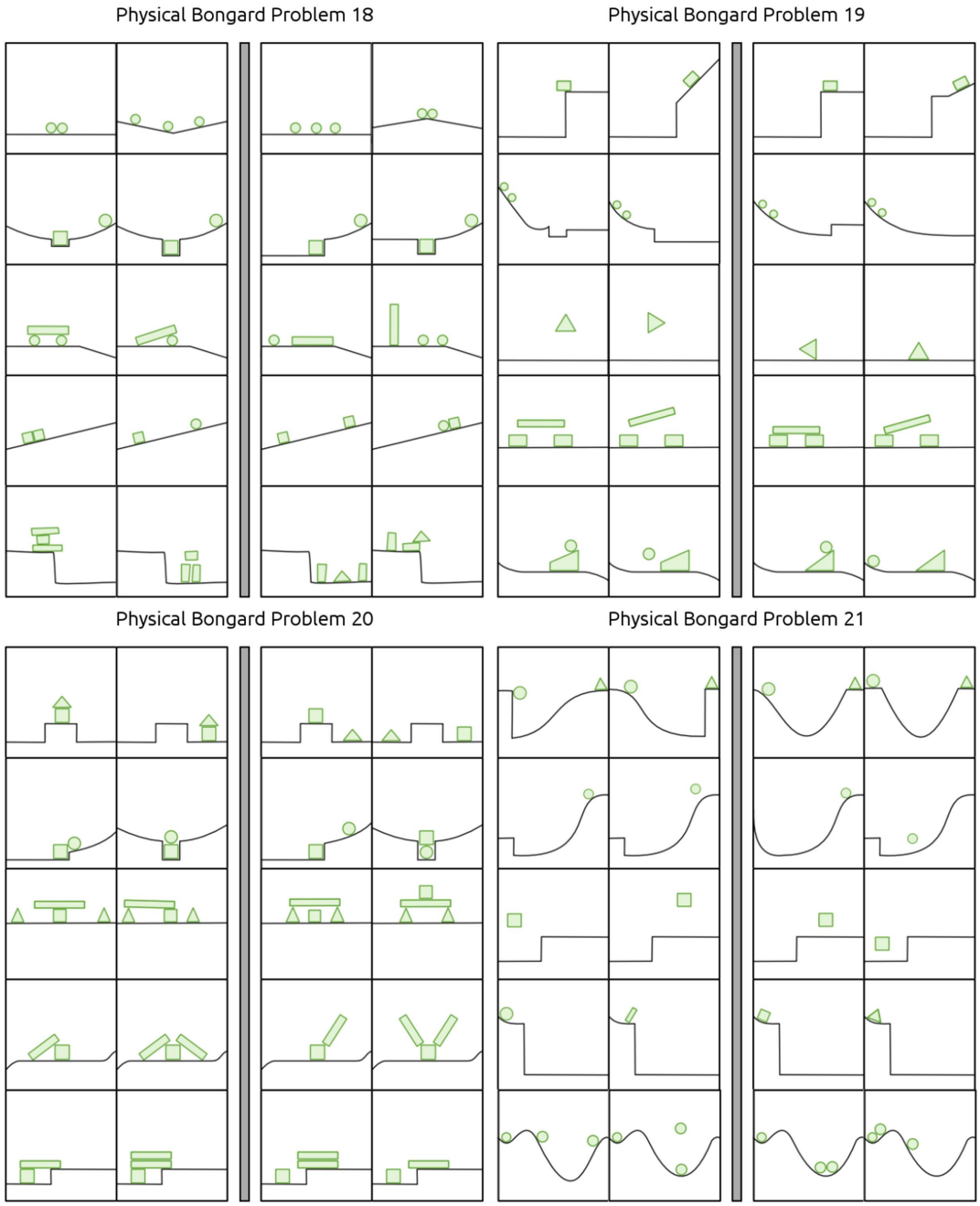
All Bongard Problems Tested on Humans and PATHS



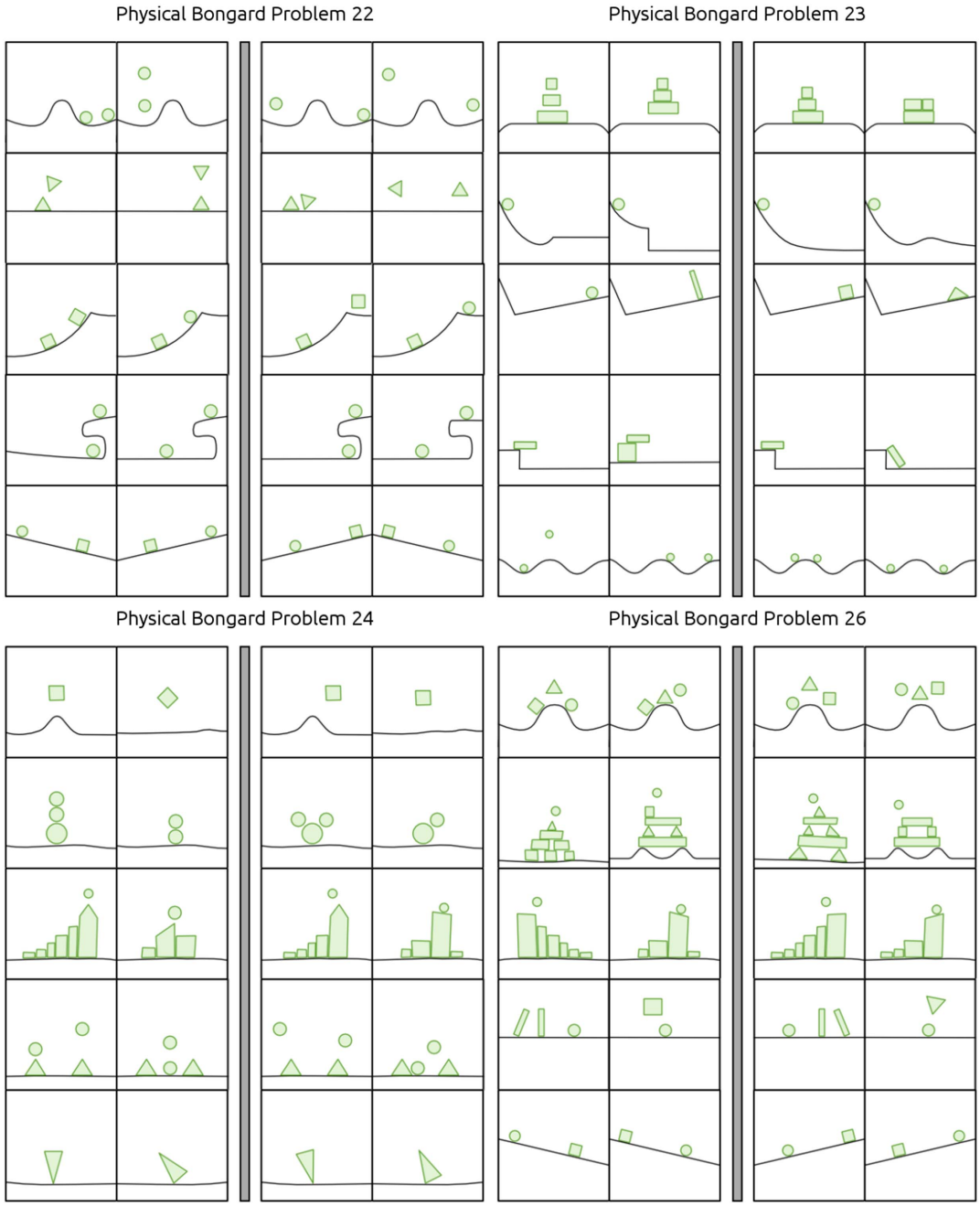
(*Appendices continue*)



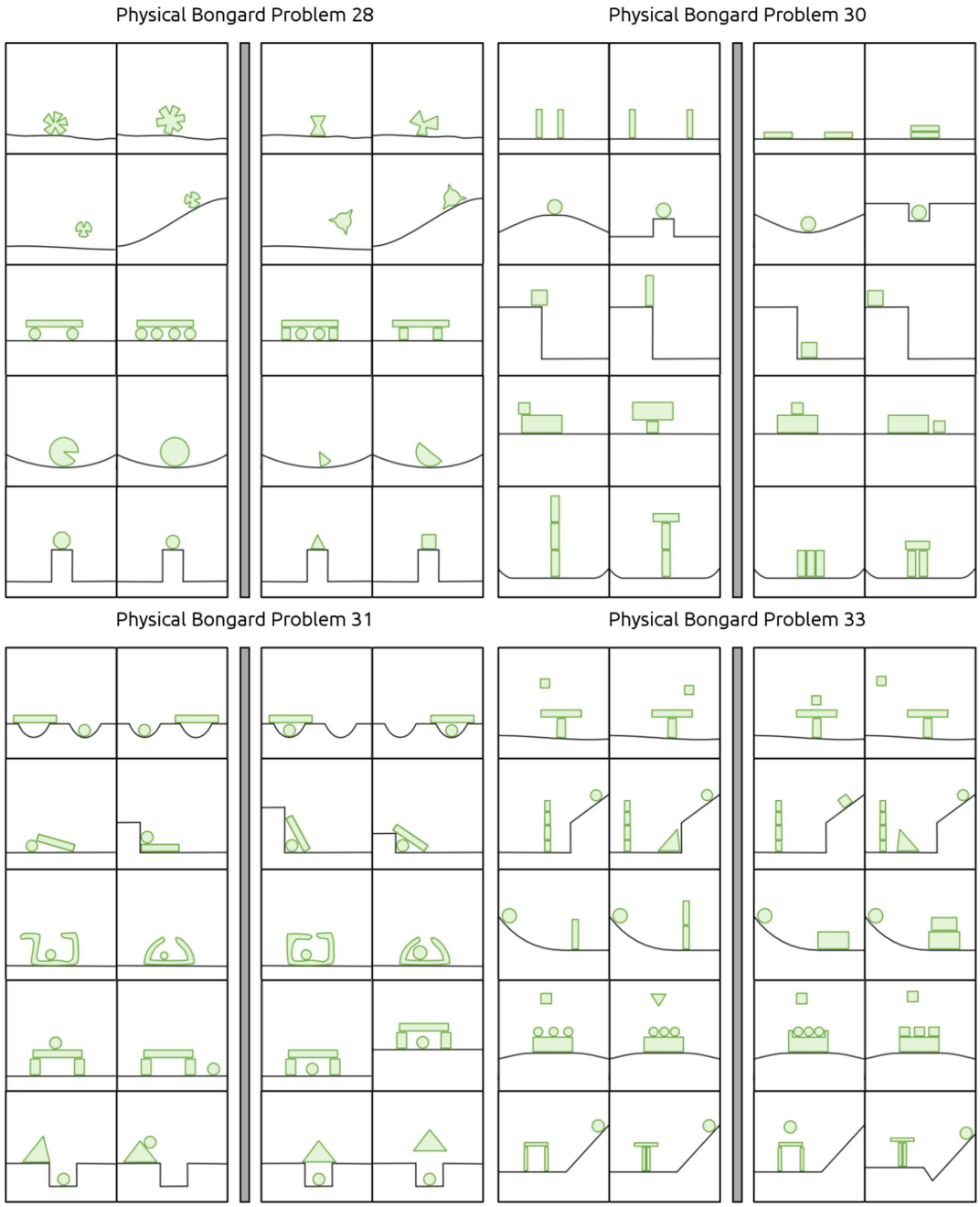
(*Appendices continue*)



(*Appendices continue*)



(*Appendices continue*)



Note. PATHS = perceiving and testing hypotheses on structures. See the online article for the color version of this figure.

(*Appendices continue*)

Appendix C

Solutions to All Bongard Problems in Appendix B

PBP	Left side	Right side
02	One object	Two objects
04	Squares	Circles
08	Unstable situation	Stable situation
09	Objects move in opposite directions	Objects move in same direction
11	Objects close to each other	Objects far from each other
12	Small object falls off	Small object stays on top
13	Objects form a tower	Objects form an arc
16	The circle is left of the square	The square is left of the circle
18	Objects touch eventually	Objects do not eventually touch
19	An object flies through the air	All objects always touch something
20	Square supports other object	Square does not support other object
21	Strong collision	Weak or no collision
22	Objects collide	Objects do not collide
23	Collision	No collision
24	Several possible outcomes	One possible outcome
26	Circle moves right	Circle moves left
28	Rolls well	Does not roll well
30	Unstable situation	Stable situation
31	Circle can be picked up	Circle cannot be picked up
33	Construction gets destroyed	Construction stays intact

Note. PBPs = physical Bongard problems.

Received March 29, 2021
Revision received April 13, 2023
Accepted April 30, 2023 ■