Annual Review of Developmental Psychology

Bayesian Models of Conceptual Development: Learning as Building Models of the World

Tomer D. Ullman[1,3] and Joshua B. Tenenbaum[2,3]

[1]Department of Psychology, Harvard University, Cambridge, Massachusetts 02138, USA

[2]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA; email: jbt@mit.edu

[3]Center for Brains, Minds and Machines, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Annu. Rev. Dev. Psychol. 2020. 2:533–58

First published as a Review in Advance on October 26, 2020

The *Annual Review of Developmental Psychology* is online at devpsych.annualreviews.org

https://doi.org/10.1146/annurev-devpsych-121318-084833

Keywords

cognitive development, Bayesian models, intuitive theories, program learning, computational modeling, child as hacker, core knowledge

Abstract

A Bayesian framework helps address, in computational terms, what knowledge children start with and how they construct and adapt models of the world during childhood. Within this framework, inference over hierarchies of probabilistic generative programs in particular offers a normative and descriptive account of children's model building. We consider two classic settings in which cognitive development has been framed as model building: (*a*) core knowledge in infancy and (*b*) the child as scientist. We interpret learning in both of these settings as resource-constrained, hierarchical Bayesian program induction with different primitives and constraints. We examine what mechanisms children could use to meet the algorithmic challenges of navigating large spaces of potential models, in particular the proposal of the child as hacker and how it might be realized by drawing on recent computational advances. We also discuss prospects for a unifying account of model building across scientific theories and intuitive theories, and in biological and cultural evolution more generally.

Contents

1. INTRODUCTION

The central cognitive task of early childhood is to build a model of the world. The chief aim of computational approaches to cognitive development is to build models of this task and of children's minds as they solve it.

This enterprise sits necessarily at the intersection of two fields. Researchers in cognitive development ask the big questions: What knowledge is there from the beginning? How do we learn the rest? What forms does our knowledge of the world take? How does knowledge change over time, and are these changes gradual modifications or wholesale transformations? Researchers in artificial intelligence (AI) and machine learning try to answer these questions normatively: What knowledge should we build into our systems? What should their initial architecture be? How should these systems learn the rest?

We see ourselves as members of both the cognitive development and computational modeling tribes, telling our fellows about converging steps these groups have recently taken toward answering these questions by using a Bayesian approach that has been influential in both disciplines and that we ourselves have contributed to. We think both sides can profit from such an exchange. In this we are empiricists: It has happened several times before.

We focus on the tool kit of hierarchical Bayesian models (HBMs) defined over structured representations, especially probabilistic generative programs that provide compelling ways to express, learn, and reason about knowledge that is abstract, causal, and generalizable. We structure our review into four main parts, addressing four specific ways these tools can help answer the big questions. The framework offers a precise but general formulation of the learning challenge: What does it mean to build a model of the world from experience, what is the logic by which this problem can be solved, and what are the possible forms our models might take? We thus begin in Section 2 with a high-level overview of hierarchical Bayes and probabilistic programs, and examples of how these tools have been used to formalize human learning as inference to the programs most likely to explain what we observe. We then turn to cognitive development more properly. Section 3 examines what initial knowledge is needed to get model building off the ground in infancy and explores the suggestion that core knowledge of objects, agents, space, and time can be formalized as a start-up library of probabilistic generative programs built by evolution. Section 4 asks how model building proceeds beyond this starting state, focusing on learning from the standpoint of the child as scientist. We interpret theory learning as the search for generative programs occurring at different timescales under different constraints from the evolutionary process that led to core knowledge. We also consider the algorithmic challenge of searching through vast spaces of candidate theories as well as mechanisms children might use to navigate these spaces: learning as stochastic search and learning as programming (or the child as hacker). In Section 5 we move from the child as scientist to the scientist as scientist, and discuss how the Bayesian approach helps us understand both the deep similarities and the fundamental tensions between these two modes of model building: If cognitive development is like science, why is science often so hard, unreliable, and counterintuitive, while the growth of commonsense thought feels—at least in retrospect—effortless and inevitable? We close in Section 6 with a more general perspective on the activity of model building, as well as the lacunae and limitations in our own attempt to build models of it, which we hope to see addressed in future research.

2. MODEL BUILDING AS BAYESIAN INFERENCE

Informally, Bayesian inference provides the mathematics linking models of the world to the evidence that supports them. We specify a hypothesis space of possible models, and Bayes's rule determines rational degrees of belief (probabilities) for each hypothesis given some evidence. A particular model m is framed in terms of variables describing the entities in a domain, relationships between these variables, and joint probability distributions over the values of these variables and others representing the observations that could constitute evidence for the model. A model can instantiate many different forms of knowledge, from a simple parameter like the bias of a coin to a multidimensional distribution of object shapes and material properties to much more complex structures such as a causal network, the grammar of a language, or a model of intuitive physics. The probabilities inferred over model variables can be used to make predictions about future data points or to estimate functions that can be expressed in terms of the model but are not explicitly part of it, by marginalizing or integrating out model variables. For example, after seeing a coin come up heads 10 times in a row, we might use a model to infer the likely bias of the coin and then infer how likely it is to produce 10 more heads in a row the next time we flip it. Similarly, when we see an object move in a certain way during a collision, we can use a model of intuitive physics to infer its weight and thereby estimate how much force we might need to apply when picking it up.

We can illustrate the dynamics of Bayesian learning visually, by considering the set of all possible models M within a domain as defining a landscape (**Figure 1**). Each point on the landscape corresponds to a particular model m. Our belief in these models forms a probability distribution, which we visualize as the landscape's height, proportional to how likely we believe the model at each point is to be true. New evidence shifts the landscape, increasing our confidence (sharpening a peak) or altering our views (raising valleys and leveling mountains). This picture is a simplification, but a useful one.

Mathematically, Bayes's rule describes how we should move the mountains and valleys in light of observations. For a given model m and evidence e, we have

$$P(m|e) = \frac{P(e|m)P(m)}{P(e)} = \frac{P(e|m)P(m)}{\sum_{m' \in M} P(e|m')P(m')}. \qquad 1.$$

The posterior probability $P(m|e)$ is our degree of belief in model m conditioned on observing evidence e. It is proportional to the likelihood $P(e|m)$, or the probability of observing the evidence

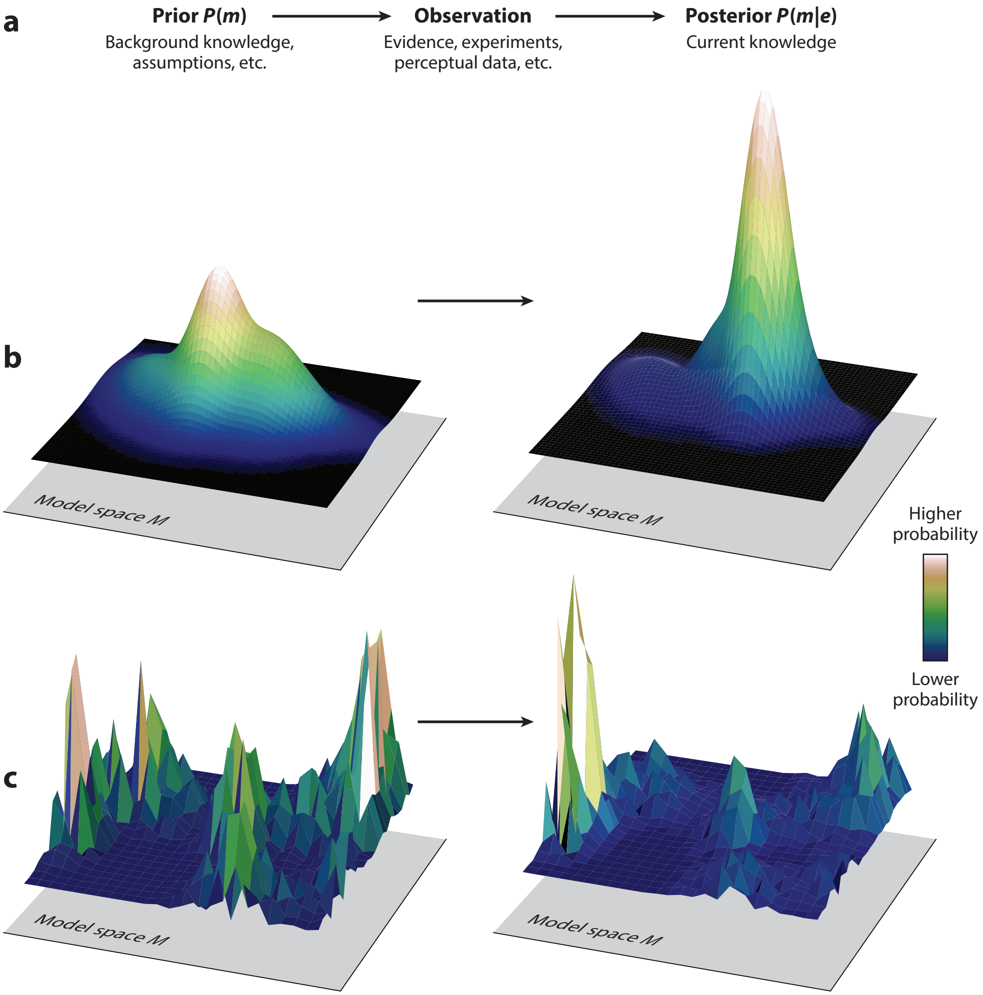


Figure 1

(*a*) The logic of Bayesian updating. The prior probability over possible models $P(m)$ shifts when new evidence e is observed, forming a posterior probability distribution $P(m|e)$. (*b*) Schematic of smooth space of models representing relatively simple parameter change. (*c*) Schematic of discrete space of models, corresponding to harder-to-search-through conceptual spaces.

given that the model is true, multiplied by the prior $P(m)$, our degree of belief in the model independent of observing e. The posterior normalizes this joint probability by dividing it by $P(e)$, the total probability of the evidence, which is simply the average of the likelihoods $P(e|m')$ from all other models m' in the hypothesis space M weighted by their priors $P(m')$. These are just the basics, presented informally. More thorough, formal treatments can found elsewhere (Gelman et al. 2013, Jaynes 2003, Russell & Norvig 2020).

2.1. Hierarchical Bayesian Models and Domain Knowledge

HBMs extend this picture to a nested tower of inferences: hypothesis spaces of hypothesis spaces and priors on priors. These overhypotheses and hyperpriors capture general beliefs that apply

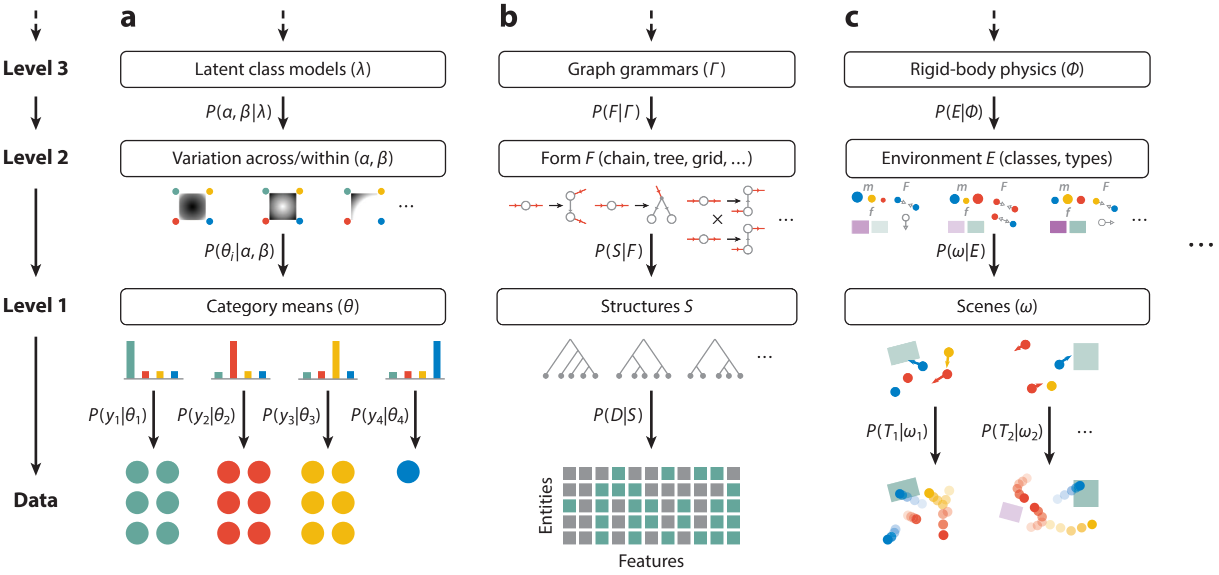

a
b
c
Level 3
Latent class models (λ)
Graph grammars (Γ)
Rigid-body physics (Φ)
$P(\alpha, \beta|\lambda)$
$P(F|\Gamma)$
$P(E|\Phi)$
Level 2
Variation across/within (α, β)
Form F (chain, tree, grid, ...)
Environment E (classes, types)
m
F
f
...
$P(\theta_i|\alpha, \beta)$
$P(S|F)$
$P(\omega|E)$
Level 1
Category means (θ)
Structures S
Scenes (ω)
$P(y_1|\theta_1)$
$P(y_2|\theta_2)$
$P(y_3|\theta_3)$
$P(y_4|\theta_4)$
$P(D|S)$
$P(T_1|\omega_1)$
$P(T_2|\omega_2)$
Data
Entities
Features

Figure 2

Examples of hierarchical generative models. Each level inherits from the level above, which defines the distribution $P(\text{Level}_i|\text{Level}_{i+1})$. Higher levels capture more abstract, domain-general knowledge. (*a*) Latent class models for discrete properties of objects (from Kemp et al. 2007), which can learn the distribution of a property within a category while acquiring a general expectation about the distribution of properties. (*b*) Graph grammars over structural forms (from Kemp & Tenenbaum 2008), which can simultaneously learn, for instance, that a tree model is better suited for describing a domain than a grid as well as the specific tree models. (*c*) Rigid-body physics programs for dynamic scenes (from Ullman et al. 2018), which can reason about objects and dynamics.

across objects or situations within a broad domain and generate the concrete hypotheses and priors needed to form models of these specific cases. Inference at higher levels of the hierarchy can explain how priors that guide future learning can themselves be learned, by integrating evidence from lower levels across specific cases previously encountered. HBMs are thus especially relevant for modeling learning in childhood (Tenenbaum et al. 2011). They can explain how children learn abstract knowledge that organizes a domain even before they work out specific details—what has been called the "blessing of abstraction" (Goodman et al. 2011). They support rapid inferences about which generalizations apply to new instances in a domain, accounting for children's ability to perform one-shot learning of new concepts as well as to learn how to learn in this way (Kemp et al. 2007). Most fundamentally, HBMs provide a general formalism for what must be built-in (the most abstract overhypotheses and primitives) and what can be learned (potentially, all lower-level constraints and hypotheses).

We now outline a simple example illustrating these points (**Figure 2*a***). For more details and further applications of HBMs in cognition and development, we refer readers to Dewar & Xu (2010), Goodman et al. (2011), Gopnik & Wellman (2012), Griffiths et al. (2008), Kemp (2008), Kemp et al. (2007), Tenenbaum et al. (2006, 2011), and especially the tutorial by Perfors et al. (2011).

Imagine you are presented with a number of bags, each containing marbles of some unknown colors, and asked to guess which color will be drawn next from a given bag (this example is based on Goodman 1983 and Kemp et al. 2007). Before you draw any marbles, your guess about the distribution of the colors in any particular bag might be fuzzy at best. Suppose you then draw a single green marble from the first bag. Your hypothesis changes somewhat in favor of a greater proportion of green marbles in the bag, but you probably would not bet a large sum that all the marbles remaining are green. Suppose you continue to draw from the bag and find that the next five marbles are all green. At this point, you might reasonably suppose the bag contains mostly or even all green marbles. That is, the posterior probability P(next marble = green|6 green marbles, background assumptions) is relatively high, much higher than the prior P(next marble = green) would have been when you started out—a perfectly straightforward case of Bayesian inference. You now move on to the other bags. You draw six marbles from the second bag and find that they are all red. From the third bag, you draw six yellow marbles. At this point, what is your guess for the color distribution of the fourth bag? It is quite reasonable to say "I don't know the specific color, but I bet all the marbles in that bag are the same color." If you then draw a single marble from the fourth bag and find that it is purple, you would probably be quite confident that the next draws from that bag would also be purple—almost as confident as you were about contents of the first four bags, after seeing many more draws from them.

These patterns of inference arise naturally from a hierarchical Bayesian analysis. Your specific guess about the color of the marbles in a bag was informed by marbles drawn from that bag (evidence informs hypotheses). But, under the assumption that the bags are produced more or less in the same way, your guess was also informed by a more general understanding of this domain of marble bags (overhypotheses shape hypotheses). This general understanding was formed from the same data you used to learn about the contents of specific bags (overhypotheses and hypotheses can be learned at the same time). And you were reasonably justified in guessing that an entire bag was filled with purple marbles from only a single purple observation, based on your understanding of the domain (overhypotheses support rapid generalization). Of course, even before seeing a single marble from a single bag, the notion that the marbles might be all the same color could have been more likely in your mind than the notion that they follow a peculiar distribution (overhypotheses are themselves shaped by more basic assumptions, either learned or built-in).

Speculating about colored marbles might seem a far cry from the problems of learning in cognitive development, but the math behind this example (a hierarchical Dirichlet-multinomial model, for mixtures of attributes in latent classes) has been used to explain how children can acquire important inductive constraints from very limited experience, such as the shape bias for artifact names in word learning (Kemp et al. 2007) or semantic constraints on syntactic alternations in early verb learning (Perfors et al. 2010). The ability to make these higher-level inferences also appears quite early, as Dewar & Xu (2010) showed in a series of elegant studies on overhypothesis learning in 9-month-old infants.

Learning overhypotheses on mixtures of latent properties is perhaps the simplest setting in which HBMs have been applied, but more complex hierarchical models can be defined over richer and more diverse representational structures (Griffiths et al. 2010, Tenenbaum et al. 2011). HBMs can even be used to infer the form of structure most appropriate for reasoning in a domain (**Figure 2*b***). For example, tree-structured representations may be particularly useful for reasoning about the names and properties of types of objects (e.g., Kemp & Tenenbaum 2009, Xu & Tenenbaum 2007), whereas directed graphs (Pearl 2000) may be more useful for capturing causal relationships between objects and their properties (Gopnik et al. 2004, Tenenbaum et al. 2007). Kemp & Tenenbaum (2008) showed how these and other forms of structured probabilistic models—including cliques, chains, grids, rings, and many asymmetric directed models—can all be generated by graph grammars, with a metagrammar that places a high-level prior on these grammatically defined model classes. Inference in the corresponding HBM would thus allow a learner to grasp, for example, that "in this domain a tree form is most useful," even before figuring out the specific tree that best organizes the objects and their properties.

These examples based on mixtures, graphs, and grammars represent only a few points in the landscape of possible ways to represent the overhypotheses that people learn about domains. HBMs have also been defined on vector spaces, relational schemas, and first-order and higher-order logic in order to capture different aspects of domain structures. But, generally, all of these representations can be regarded as species of a single unifying probabilistic representation based on programs, which we turn to in the next section.

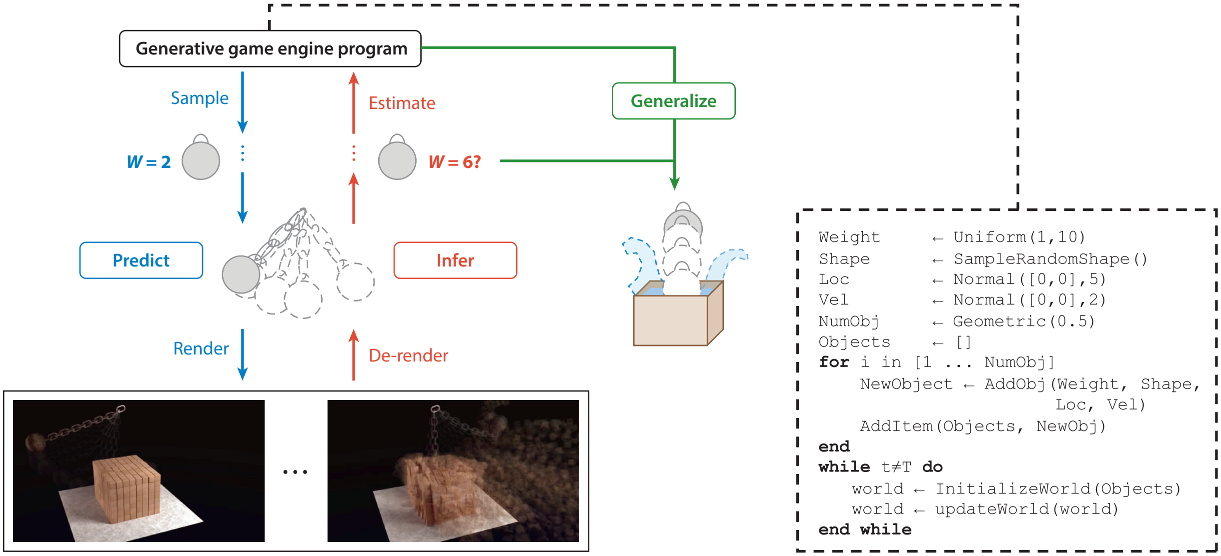
2.2. Probabilistic Generative Programs, Simulators, and Mental Models

Just as Turing machines are universal models of computation, probabilistic programs are universal probabilistic models. Their model space M comprises all computable probabilistic models, or models where the joint distribution over model variables and evidence is Turing computable. The ProbMods web book (**https://probmods.org**; see Goodman et al. 2016) provides a comprehensive introduction to probabilistic programs and their use in cognitive modeling, along with many interactive examples.

In addition to unifying all of the familiar HBMs and overhypothesis models discussed above, probabilistic programs have given Bayesian computational accounts their first viable means of addressing learning and inference with rich mental simulations of causal processes—the kinds of representations needed to capture people's mental models of physical objects, intentional agents, and their interactions in space and time. Because these concepts have been the focus of much research in cognitive development, including both infants' core knowledge and children's development of intuitive theories, we focus on probabilistic programs in this setting.

Informally, we can think of a probabilistic program as simply a computer program that makes probabilistic choices. For a given input, there is no single deterministic output; rather, there is a probability distribution over outputs. We focus on probabilistic generative programs in which the program describes processes that create possible worlds. The program specifies the types of entities in the world, their properties, their interactions, and their dynamics over time. These correspond to the variables and relations between variables in a probabilistic model, as described in Section 2.1. A single run of the program samples values for each variable, generating one possible way the world could unfold—intuitively, imagining a possible world. Some of these variables will be observable and constitute possible evidence for a learner trying to infer the generating model. Repeated samples specify a probability distribution over possible worlds, the joint distribution over model and evidence variables. Bayesian inference can be thought of as running the program backward: observing some evidence that is the partial output of the program and then trying to infer the most likely inputs and random choices made throughout the generating run that led to that evidence, thus inferring the unobservable variables that best explain the observables.

The idea that the mind builds internal models that mimic the world's causal structure, which can be used to simulate what will happen, what did happen, what could happen, and what would happen if, is one of the founding ideas of cognitive science. It predates the field (Craik 1943; see also Johnson-Laird 2004 for a more expansive history) and has shaped it since early days (Gentner & Stevens 1983). Probabilistic generative programs can be regarded as a modern incarnation of these ideas that combines abstract, structured causal knowledge with capacities for Bayesian inference and simulation-based reasoning over those models (Gerstenberg & Tenenbaum 2017). This approach has recently been applied across cognitive science, especially in concept learning (Lake et al. 2015), causal and counterfactual reasoning (Chater & Oaksford 2013, Gerstenberg & Tenenbaum 2017), object perception (Erdogan & Jacobs 2017, Wu et al. 2015), action understanding or intuitive psychology (Baker et al. 2017, Jara-Ettinger et al. 2016), and intuitive



```
Weight       ← Uniform(1,10)
Shape        ← SampleRandomShape()
Loc          ← Normal([0,0],5)
Vel          ← Normal([0,0],2)
NumObj       ← Geometric(0.5)
Objects      ← []
for i in [1 ... NumObj]
    NewObject ← AddObj(Weight, Shape,
                       Loc, Vel)
    AddItem(Objects, NewObj)
end
while t≠T do
    world ← InitializeWorld(Objects)
    world ← updateWorld(world)
end while
```

Figure 3

Simplified mental game engine program for probabilistic reasoning including prediction, inference, and generalization. (*Inset*) Part of the program pseudocode prior to conditioning (based on Battaglia et al. 2013, Gerstenberg & Tenenbaum 2017, and Ullman et al. 2018). Still images at lower left reproduced with permission from a video created by Andrew Price.

physics. Below, we go into more detail about intuitive physics, which is also the domain that has been best modeled using probabilistic programs across different stages of cognitive development.

Consider people's ability to reason about the dynamics of everyday objects—how things bounce, ooze, tumble, crash, drape, drip, or snap. People can use their intuitive physics to predict, explain, and reimagine the world. But what form does this knowledge take? One suggestion is that this knowledge is embodied in an approximate, probabilistic, generative simulator for the physical world (Battaglia et al. 2013, Hamrick et al. 2016, Sanborn et al. 2013, Ullman et al. 2017).[1]

A generative program starts from positing the world that gives rise to perception. For intuitive physics, think of something like the physics engine programs used in modern video games to create real-time interactive environments (e.g., Gregory 2018). A probabilistic physics program would similarly begin with simplified objects, properties, and dynamics that evolve the world state, which can be connected to an observation or rendering function analogous to a game-style graphics engine. **Figure 3** illustrates an example for one set of scenes (see **Figure 2*c*** for a different physical domain framed as an HBM). Each run of the program generates a physical scene unfolding over time, and the program can be thought of as defining a prior distribution over a vast space of possible sets of objects and their trajectories.

As a probabilistic model, this generative program can be used for many purposes (**Figure 3**), including the following:

1. Prediction. If we know or assume the values of key variables, such as objects' locations, shapes, weights, and velocities, we can simulate the resulting world forward to answer such

[1] Of course, this is just one potential answer. Other representations for intuitive physics include perceptual heuristics (Gilden & Proffitt 1989), qualitative process models (Forbus 2019), and neural networks (Lerer et al. 2016).

questions as: Will this swinging ball hit these blocks? Where will this block end up? We can also predict counterfactually: What would have happened if the ball had been much lighter than the blocks?

2. Inference. Given an observed trajectory, we can find the distribution over the values that led to it, answering such questions as: If the ball bounced off the blocks without moving them, how heavy are the blocks? What material are they made of? Inference can happen at multiple levels of a hierarchical model (**Figure 2*c***), ranging from simple parameter estimation (how heavy a block or the ball is) to structural relations (the ball appears to be attached to the chain) to higher-level concept discovery (an unknown force appears to hold the blocks together).
3. Generalization. Knowledge gained from one observed setting carries over to others. Having learned from collisions that the ball is heavy, we can modify parts of the program to predict how it will behave in a new setting, such as being dropped into a tub of water.

Beyond any specific use, such a generating program is also a highly compact representation. Compare the small number of parameters (object positions, shapes, etc.) needed to create the dynamic scene in **Figure 3** with the millions of numbers needed to recreate it pixel by pixel, frame by frame. The generative program's description is far more compact, even taking account of the cost of the full physics simulator, once sufficient data are observed.

Thinking in terms of compression is also helpful for understanding how a physics simulator, or any probabilistic generative program, could be learned. Probabilistic programs let us consider model building in its most general form as a Bayesian inference. In Equation 1, the model space M now spans all possible generative programs. We seek the program m that best explains our observations—that maximizes the posterior probability $P(m|e)$, where the likelihood $P(e|m)$ is now the probability of the observations e being generated by program m, and the prior $P(m)$ can be derived from the program's description length (Li & Vitányi 2008) in a probabilistic programming language. Equivalently, that language could also be expressed as a probabilistic generative model, equipped with a probabilistic grammar. Model building most generally is simply inference in a hierarchical probabilistic program—a program-generating program.

With this full picture in place, we can now return to the central questions of cognitive development. We rightly start at the beginning, with the question of what knowledge is there earliest in life.

3. CORE KNOWLEDGE AS START-UP SOFTWARE

Within the hierarchical Bayesian program-learning framework, the question of what knowledge is there at the beginning becomes: What is the start-up library of program components—functions, variables, routines, and complete programs, as well as program-learning mechanisms—that is available in the child's mind, independent of their experience in the world?

Before turning to actual children, it is helpful to examine this question in a normative sense. Imagine yourself as a modern-day Mister Geppetto trying to build a machine child. How would you go about it? When Alan Turing considered this issue, he presumed that "the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets" (Turing 2009, p. 60). In other words, start without specific program content, only with a general ability to learn programs, and then consider all possible programs that could account for one's experience. While elegant in its simplicity, such an approach comes with a monstrous search problem. Even optimal program search (Levin 1973) would still take ages to find programs accounting for even relatively simple inputs—let alone to discover something like a physics engine program from scratch. It is also needlessly inefficient: Unless this is the first machine of its kind, why shouldn't it benefit from the discoveries of its ancestors? So let Geppetto build in a start-up library of useful computational primitives—much as evolution provides to every organism.

It would be easy to lose balance and overcorrect, building a creature from only rigid, thoughtless routines. A centipede might instantiate such a rigid design, one that hardly adapts during the creature's lifetime. This is not necessarily a bad idea: If a centipede program produces an optimal way to behave in a centipede niche, one might as well preload a centipede with it, rather than it having to learn to become a centipede anew each generation. But this is not the way to build a machine that is intelligent in any humanlike fashion.

In between these extremes are many possibilities, but the most promising, we believe, is minimal nativism. This could also be what Turing (2009, p. 62) had in mind when he hypothesized that "[o]ne might try to make [the child machine] as simple as possible consistently with the general principles." These principles start with learning mechanisms: In the space of possible programs are programs that can learn other programs. The learning process for creatures that carry out such programs is different from that for evolution, as these creatures can iteratively build and refine their models on the basis of their own experiences. One would expect such creatures to have a start-up library, but one that is more flexible, general, and abstract than rigid, contentless behavioral routines (Baum 2004, Lake et al. 2017).

3.1. Domains and Principles of Core Knowledge

Several decades of infant research have offered empirical evidence for the existence of just such a start-up library in the human mind (e.g., Carey & Spelke 1994; Spelke 1990, 1992, 1995; Spelke & Kinzler 2007). Core knowledge refers to early-emerging, possibly innate expectations that infants have about the world. As one might expect from a conserved start-up library of programs, these expectations are cross-cultural, shared to some extent with many other species, and confined to a minimal set of domains that would be relevant in almost any situation, such as objects, agents, space, time, number, and social relations.

The principles of core knowledge are general and abstract, applying to every entity in a domain. For example, take the principle of solidity in core physics (Baillargeon et al. 1985, Stahl & Feigenson 2015): Even young infants expect that a rigid body in motion will not pass through another solid body like a ghostly apparition. This principle applies to every physical object. It does not have to be learned anew each time for cars, then jars, then guitars. Or consider the principles of permanence and continuity: Young infants expect objects to continue to exist behind occluding barriers and are surprised if an object seems to magically teleport or disappear (Spelke et al. 1995). Again, this expectation does not depend on an object's shape, color, or texture; what matters is that it is an object. Core knowledge principles are also domain specific. Knowing that something is an agent does not entail solidity or continuity; after all, we can believe in ghosts. But all agents—ghosts included—respect core principles of goal-directedness and efficiency (e.g., Csibra 2008; Csibra et al. 2003; Liu et al. 2017; Woodward 1998, 1999).

3.2. Core Knowledge Principles as Generative Programs

What kind of knowledge is core knowledge? Core principles such as "solid bodies cannot move through one another" are hard-won, deeply explanatory, scientific generalizations. But they are still just statements in a human language. We as adults grasp their meaning because we understand language, but that does not translate into a form that can be implemented computationally, or in the mind of an infant who has not learned language. Here is where the tools we have developed become useful. We propose that core knowledge can be thought of as built-in, minimal, domain-specific libraries of generative programs. These programs implement the expectations of core knowledge through probabilistic simulation, which offers a different and valuable way to think about the form that core knowledge takes.

Traditionally, core knowledge has been expressed in terms of abstract principles, without specifying how or whether these principles are represented explicitly in infants' minds or what kind of explicit or implicit reasoning takes the infant from abstract principles to concrete expectations. A probabilistic simulation framework makes these commitments much clearer and suggests how different core knowledge principles might be implemented differently: some as explicit constraints enforced by modifying simulations that violate them, others as implicit constraints implemented in the logic of how the simulator updates over time, and still others as merely emergent statistical properties of how simulations tend to unfold. As an illustration, consider a minimal game-style physics engine with the capacity to represent only coarse object shapes with basic rigid-body dynamics. The principle of solidity is explicitly implemented in typical physics engines via a function for collision resolution, which checks at each time step whether objects are about to move into overlapping regions of space and modifies their trajectories, if necessary (Gregory 2018). The principles of continuity and permanence, by contrast, are implemented implicitly via the simulator's dynamics: The position and velocity of each object in the simulation are updated according to rules of dynamics, such as Newton's second law, $a = F/m$, after computing the sum of forces F incident on it. These update rules ensure that objects in motion change their positions only locally from one time step to the next for any reasonable F (continuity) and that each object present at a given time continues to exist at all future times, staying in its current location if it is stably supported and not moving and if no object makes contact with it (permanence). There need not be any line of code explicitly stipulating "thou shalt not make objects disappear," unlike the function explicitly prohibiting solid objects from interpenetrating. Still other principles, such as "sand accumulates in piles" (Anderson et al. 2018), merely arise from the dynamics in particle-based simulations, which have been used to model people's intuitions about nonsolid substances (Bates et al. 2019, Kubricht et al. 2017), without the need for any explicit or implicit notion of piles. They reflect the emergent behavior of particles in the simulator in aggregate, much as piles of sand are emergent phenomena of actual sand particles in the real world (Ullman et al. 2017).

This approach to core knowledge has been implemented in working computational models. Smith et al. (2019) show that a game engine with simplified dynamics and coarse object representations, combined with probabilistic inference to track objects behind occluders, can account for many basic physical expectations shown in 4-month-old infants. The same model can predict adults' quantitative judgments about how surprising a scene is, when the surprise occurs, and what kind of violation likely occurred (Smith et al. 2020). A similar coarse probabilistic model predicts 12-month-olds' looking times for a range of multiobject dynamic displays, varying in how much they violate continuity (Téglás et al. 2011).

Learning within core generative programs is limited, but possible, and accounts for some of the learning that happens over the first year of life. Consider, again, the program shown in **Figure 3**. Such a program could start out uncertain about the weight of an object and then, after observing a collision between it and a second object, become more certain about the relative weights of the two objects. This is a very limited sort of learning. But the program can also place probabilities over hypotheses at higher levels of abstraction, such as the types of properties objects have or the way the force laws work, while keeping the fundamental structure the same (e.g., Ullman et al. 2018). This would correspond to allowing inference at higher levels of the HBM shown in **Figure 2**. To anthropomorphize: The program would say, in effect, "I know that there are objects, and they have shapes and weight and other properties, and there are forces in the world that update the properties of the objects moment to moment, but I have very little idea what those forces and properties and shapes are exactly." Learning over the first year of life, for example, the emerging understanding of gravity and stability or the relationship between weight and size (e.g., Baillargeon et al. 2008, 2011) would thus be learning within the limits of a domain-specific physics program (Ullman et al. 2017, Wu et al. 2015).

Analogous probabilistic generative programs can be constructed for other core domains. Core psychology has been especially well captured by modeling agents as approximate utility maximizers, choosing plans to maximize the rewards of goals minus the costs of actions (Baker et al. 2009, Jara-Ettinger et al. 2016). These models naturally embody the previously proposed principles of goal-directedness and efficiency, but, embedded in probabilistic programs, they become computationally precise and able to account quantitatively for many expectations of infants and young children (Kiley Hamlin et al. 2013, Liu et al. 2017, Lucas et al. 2014).

Finally, a full account of core knowledge as a start-up library of programs needs to explain how such knowledge could be discovered and encoded by evolution. Genetic programming methods are a promising computational direction (e.g., Czégel et al. 2018, Koza et al. 1994, Stanley et al. 2019). Applied to search for probabilistic generative programs, they could provide at least a first candidate hypothesis for how core knowledge was originally constructed.

4. CHILD AS SCIENTIST, CHILD AS PROGRAM LEARNER

4.1. Intuitive Theories

In addition to their evolutionary endowment and learning within the constraints of core systems, children and adults can build genuinely new models of the world around them. These models have been likened to the theories that scientists build, under the banner of the "theory theory" and the child as scientist. The analogy addresses both the forms that knowledge takes and the ways in which evidence is gathered and evaluated (e.g., Carey 1985, 2009; Gopnik & Meltzoff 1997; Gopnik & Wellman 1994; Lombrozo 2016; Murphy & Medin 1985; Schulz 2012b; Wellman & Gelman 1998). Like scientific theories, intuitive theories posit the existence of classes of underlying variables and causal laws relating them. Also like scientific theories, intuitive theories are generalizations from observed data and are subject to revision given new evidence. They are evaluated on the basis of predictive power as well as other explanatory virtues, such as simplicity, coherence, and generality.

As with proposals for core knowledge, intuitive theories were originally framed as informal accounts in natural language. However, since the late 2000s, HBMs and probabilistic programs have been extensively developed to capture the structure of intuitive theories and how they can be learned (Gerstenberg & Tenenbaum 2017, Goodman et al. 2014, Gopnik & Wellman 2012, Tenenbaum et al. 2011). These efforts not only build on but also, importantly, go beyond earlier models of causal learning in children and adults (Gopnik et al. 2004, Griffiths & Tenenbaum 2005) defined on causal Bayesian networks, or directed graphical models (Pearl 1988, 2000). Hierarchical models are needed to capture the different levels of abstraction in intuitive theories—high-level framework theories spanning entire domains, as well as specific theories of particular causal systems within that domain (Goodman et al. 2011, Kemp et al. 2010, Tenenbaum et al. 2007). Generative programs are needed to capture the full texture of the causal processes at work—object-centric, force-based interactions in physics; goal-directed planning and perception-driven belief formation in agents; and the dynamic growth processes of living forms—none of which can be expressed simply in terms of a directed causal graph (Goodman et al. 2014, Griffiths et al. 2010).

The relationship between core knowledge and intuitive theories, between learning in infancy and later in childhood, has been subject to intense debate. Some researchers see intuitive theories and core knowledge as essentially different: Whereas theories are based on data and subject to change and refutation, core knowledge is inborn and fixed (Carey 2009, Carey & Spelke 1996). Intuitive theories, like scientific theories, are best described as systems of concepts in a "language of thought" (Fodor 1975), whereas core knowledge is made up of protoconceptual primitives (Xu 2019). In contrast, others have argued for continuity, or "theories all the way down" (Gopnik 1996, p. 510; see also Gopnik & Wellman 2012, Woodward & Needham 2008): Infants' models of the world are generalizations from data, subject to revision just as much as later-developing theories are—and, indeed, through learning from experience, evolve into those theories.

The Bayesian framework can help find common ground, clarifying both similarities and differences in how these different forms of knowledge operate (see Xu 2019 for an important and related perspective). In our view, the content of intuitive theories in childhood and that of core knowledge in infancy share key structural features: Both can be represented as hierarchies of probabilistic generative programs. But the evolutionary origins of core knowledge mean that the initial programs may be expressed in a different, more modular form, with primitive functions, variables, and data types not subject to drastic revision in the sense of a complete code rewrite. Learning is possible within core knowledge programs: for instance, learning about new forces or object properties in a mental physics engine or new overhypotheses over utility functions in a mental planning engine. Moreover, core representations may powerfully scaffold the learning of later-developing theories: The fact that core knowledge programs persist over development, even while intuitive theories come and go, allows their component variables, functions, and data structures to provide building blocks for new theories as well as crucial constraints on (or priors for) the forms that those theories will take. But this does not mean that core programs support such radical transformations in knowledge as constructing whole new programs or libraries of programs—what a new intuitive theory might achieve by drawing on domain-general languages for model building (Griffiths & Tenenbaum 2009, Kemp et al. 2008, Tenenbaum et al. 2011, Ullman et al. 2012) or a probabilistic language of thought (Goodman et al. 2014, Piantadosi et al. 2016).

4.2. Learning as Searching the Space of Programs

The picture so far—children building models of the world by updating a posterior distribution over hierarchies of generative programs—exists at a functional, ideal level (Anderson 1990, Marr 1982, Tenenbaum et al. 2011). But such a picture is in danger of being ripped apart conceptually and empirically when it comes into contact with real learners. Conceptually, if "learning" is merely shifting around probability mass, then learning is not actually happening (Fodor 1998). The ideal picture brings to mind a Newton already possessing a theory of mechanics before sitting under the apple tree, and merely updating his belief in the theory in light of the apple falling down. Also, the space of possible programs is very, very large and untamed. The posterior probability landscape over a space of discrete programs tends to look like **Figure 1*c***, with no nice structure to it—even with the constraints and priors provided by core knowledge and the form of any programming language of thought. To make matters worse, children have limited memory capacity and cannot hold and manipulate more than a tiny region of such a space in their heads. Empirically, while different children eventually tend to converge on the same intuitive theories (e.g., Carey 2009, Gopnik & Meltzoff 1997, Wellman et al. 2011), seen from close up, the process contains a lot of fizz and foam. Local theory change can seem random, with haphazard changes and even backtracks in knowledge (e.g., Siegler 2007, Siegler & Chen 1998).

The ideal picture meets these challenges by moving to the algorithmic level of analysis (Griffiths et al. 2015, Marr 1982), where it makes contact with creatures that have finite time, energy, and computational resources. Building on classic notions of bounded rationality (Simon

1956), resource-rational frameworks for approximate Bayesian learning (Gershman et al. 2015, Griffiths et al. 2015, Lieder & Griffiths 2020) implement rational approximations to the ideal-level computations under given constraints, trading off accuracy and speed of learning, or speed and memory costs, and so on. Different algorithms for searching the space of possible theories make these trade-offs differently, but the very introduction of the algorithmic view already helps address the conceptual challenge: While one can think of the space of all possible programs that a child's language of thought can express as existing in some sense, one should not think of that entire set as existing in the child's head. Rather, the resource-rational child will hold in mind a limited number of hypotheses at a given moment—maybe only one (Vul et al. 2014), as if occupying a single point in conceptual space. She then applies program transformations to her models, moving her to new positions in that space. That the combination of programs and program transformations theoretically expresses an infinite space does not mean the child doesn't learn or discover something new, in the same way that a child's ability to express and understand English does not mean that Shakespeare didn't come up with something novel when he first wrote "O brave new world, that has such people in 't!"

The suggestion that children change their world models by something like a program transformation is not itself a new idea (e.g., Simon 1962). But the rise of probabilistic generative models that can express intuitive theories also comes with new tools and metaphors for understanding the way children and adults might search the space of possible models. We consider two here, learning as stochastic search or hypothesis sampling and learning as a kind of programming (the child as coder or the child as hacker).

4.3. Theory Learning as Stochastic Search

One important class of algorithms for approximating Bayesian learning is based on stochastic (or intrinsically random) algorithms that search for hypotheses with high posterior probability, or that attempt to sample multiple hypotheses from the posterior. The broad family of Markov chain Monte Carlo (MCMC) methods in statistics and AI (including Metropolis–Hastings and Gibbs sampling) (Russell & Norvig 2020) has been frequently mined as inspiration for algorithmic-level Bayesian models of cognition (Gershman et al. 2015; Goodman et al. 2008; Griffiths et al. 2008, 2012) and is especially natural for learning probabilistic generative programs in a hierarchical Bayesian framework (Saad et al. 2019; Ullman et al. 2012, 2018).

In conceptual terms, MCMC algorithms split the search for good theories into an iterated sequence of two distinct stochastic tasks: propose and evaluate. New theories are proposed either by sampling them from a prior distribution over possible theories or by randomly sampling local changes to the currently held theory. A newly proposed theory is evaluated in relative Bayesian terms: How much better or worse does it combine explaining observations (likelihood) with plausibility and simplicity (priors), relative to the currently held theory? Proposals are accepted or rejected with some varying probability but always such that higher-scoring proposals are more likely to be accepted. Sampling thus carries out a local trek through the space of possible theories, a biased random walk that is unpredictable from moment to moment but over time is guaranteed to converge to sampling good (high-posterior-probability) theories.

Such a dynamic has been proposed to account for the characteristic dynamics of children's learning: Different children learning the same domain may take different trajectories, even in the face of the same evidence, but nonetheless converge on similar—and similarly veridical—explanations of the world (Bonawitz et al. 2019, Piantadosi et al. 2012, Ullman et al. 2012). The mathematical framework of MCMC (MacKay 2003, Russell & Norvig 2002) guarantees that different learners who share the same hypothesis space and distribution of evidence, and who all implement some MCMC algorithm, will all eventually converge to the same region of high-posterior-probability theories, even if they start in different regions of the hypothesis space or use different MCMC algorithms. If those high-probability theories are reasonably similar in form, this will seem like convergence to a shared understanding of the world. But in settings where the hypothesis space or evidence is not shared or is changing over time, or the available evidence greatly underdetermines the form of the most probable theory, such convergence may be unlikely or even impossible—a point we return to in Section 5.

The dynamics of how stochastic search evolves over time has also suggested intriguing parallels with the larger-scale arc of children's conceptual development. These algorithms often include a temperature parameter (Geman & Geman 1984, Kirkpatrick et al. 1983), starting in a "hot" regime in which unlikely proposals (theories) that may be worse than the currently held belief can still be acceptable, and gradually lowering over time in a process known as simulated annealing. Early on, transitions between theories are most random—noisy, large, and often suboptimal—but as the search cools, learners become less likely to change their theories, and especially to make large or suboptimal changes. Thinking becomes more predictable, and if the search cools slowly enough, it will almost surely converge on exactly the small set of theories with highest posterior probability (Geman & Geman 1984). A similar dynamic occurs in an online learning setting, even without an explicitly varying temperature parameter: As more data are encountered, the posterior over theories sharpens (**Figure 1**), which effectively results in lower-temperature search. Might children's thinking also evolve like this, for analogous functional reasons (Gopnik et al. 2015, 2017; Ullman 2015; Ullman et al. 2012)? Whether through an explicit temperature parameter and an annealing schedule that injects more randomness into younger children's thoughts and actions or simply through an implicit annealing dynamic that emerges from online learning with increasing experience, the suggestion is that early childhood is a time of high-temperature search, full of wild variation that can be creative and useful but also random and odd, gradually cooling into adulthood's more staid but stable and successful systems of knowledge.

4.4. The Child as Coder, Hacker, Software Engineer

Even with the capacity of stochastic search to approximate ideal Bayesian learning, these algorithms often feel hopelessly inefficient. They are stumbling about aimlessly, and in that sense very unlike children, who may be stumbling about but with considerably more purpose (Chu & Schulz 2020, Schulz 2012a). To paraphrase Schulz (2012a), when children are asked questions such as "Why don't clouds fall down?" they may come up with all kinds of wrong explanations, such as "The sun is holding the clouds up" or "The raindrops inside the clouds are jumping up and down." But these proposals are at least relevant, systematically related to the phenomena to be explained, and potentially correct explanations. In contrast, an answer such as "The clouds are bigger than ladybugs" or "You can eat raw cheese" would not be false, but it would not occur to a child in this context (except as a joke or miscommunication) because it is simply irrelevant to the question at hand. The point here goes beyond the finding that children can detect irrelevant explanations (Johnston et al. 2019). Rather, the suggestion is that children are able to avoid proposing irrelevant explanations in the first place, thereby making the problem of theory search far more constrained and tractable (Schulz 2012a).[2]

[2] What makes a hypothesis relevant depends in part on children's developing understanding of a domain; in the picture of our HBMs (**Figure 2**), theories at higher levels of abstraction impose relevance constraints on hypotheses at lower levels. But domain-general constraints on well-formed thoughts are also at work. For

It remains an open question how a computational system can propose only relevant hypotheses without first proposing from a much larger set and then checking for relevance, although a number of possibilities have recently been put forward. For example, model-free learning (Phillips et al. 2019, Sutton & Barto 2018) and amortized inference, or learned data-driven proposals (Gershman & Goodman 2014, Shi et al. 2010), could be useful for populating an effective hypothesis space, conditioned on patterns in observable data. Constructing an ad hoc generative model for hypotheses on the fly, by recombining parts of recalled relevant concepts, is another possibility (Ullman et al. 2016).

Perhaps the most intriguing proposals, however, are inspired by the fundamentally goal-directed nature of thinking and learning: In searching for new models or theories, we generate certain hypotheses only because we have a sense of which thoughts—if true—could accomplish our explanatory goals (Schulz 2012a). And when we combine the centrality of goals with the notion that learning is fundamentally a search for programs, where better to look for inspiration about learning algorithms than the consummately goal-directed activity of programming? That is, learning may be best understood as a natural form of programming: constructing programs that serve a purpose or modifying programs to make them better. Learning algorithms then become programs that write programs, specifically, probabilistic generative programs in the case of learning intuitive theories. This view could help explain how children are able to come up with such rich mental models of the world so much more efficiently than any random search or reinforcement learning procedure could.

Rule et al. (2020) develop this idea under the phrase "the child as hacker." Although the child as coder/programmer/software engineer would work almost as well, Rule and colleagues favor the child as hacker because of its reference (much older than any nefarious connotations for hacking) to the most creative, intrinsically motivated modes of programming, reminiscent of the styles of learning we associate most with childhood. Rule et al. show how this analogy opens up powerful new ways of modeling both the algorithms and the goals of learning.

First, consider the goals. The hacker's ultimate goal is making her code better. But there are many different dimensions of value that matter for good programs, many different goals she could aim for proximally, and each of these corresponds to a value that could guide learners in improving their world models. The familiar epistemic virtues of Bayesian learning are included here: One can improve code by making it more accurate (higher likelihood), simpler or more compact (higher prior), or more general and robust (higher probability under a hierarchical model). But one could also aim to make code faster or more efficient—the virtues targeted in resource-rational, approximately Bayesian learning (Lieder & Griffiths 2020). Other, more aesthetic or communicative goals should be in play, too: making code more reusable, elegant, understandable, or clever, or simply more fun. Children's learning could be motivated by all these goals, which means we need learning algorithms that can optimize for all these objectives.

Now, consider those algorithms. Hackers deploy a diverse tool kit of practices and processes for making code better in all the senses described above. A few of these processes are analogous to familiar learning algorithms. For instance, we routinely optimize the performance of a system by tuning parameters in existing programs, without writing any new code. If we are tuning to improve accuracy, this is analogous to learning by gradient descent in a neural network or estimating parameters in an HBM with fixed structure. But most ways to improve programs require writing new code. This includes not only writing new functions but also extending or debugging

instance, the very use of "why" in the question "Why don't clouds fall down?" suggests that an explanation based on causal models would be relevant, while a purely descriptive statement would not be (Schulz 2012a).

old ones; rewriting or refactoring code so it becomes more efficient, understandable, or reusable; writing libraries of functions to capture frequently used procedures in a domain; and even writing whole new programming languages, which might allow new kinds of concepts or new ways of thinking to be effectively expressed. All of these more creative, structure-generating aspects of programming have parallels in children's learning, too, in the active model-building processes of analogy (Gentner 1989), bootstrapping (Carey 2009), hypothetical and counterfactual reasoning, and other modes of "learning by thinking" (Lombrozo 2016).

The view of the child as hacker/coder/software engineer also fits naturally with proposals that children draw on diverse input sources in building their intuitive theories. Children learn not only from their own observations but also by building the concepts needed to understand and produce natural language (Gopnik & Meltzoff 1997, Landau & Gleitman 1985, Waxman & Markow 1995), and then using language to build their knowledge through cultural or social means (Carey 2009, Gelman 2009, Wellman 2014). Good programming likewise draws heavily from linguistic, cultural, and social inputs. Programmers make their code better through commenting it in natural language and explaining their code to others in the essential software-development process of code reviews. And much of the code that goes into any complex software system is heavily based on code produced for other purposes, by other people. Over multiple cycles, and across many projects proceeding in parallel, this cultural evolutionary process of code sharing and adaptation can lead to rapid developments that no single coder would bring about on their own. Incorporating analogs of all these processes may prove essential in models that faithfully capture how people construct their intuitive theories.

How close are we to capturing such a rich space of programming methods and goals in a probabilistic program-learning program? That is, how close are we to actually implementing the child as hacker hypothesis in a working model? We are very far, just as we are far from having computers that can program themselves more generally. But there is research aiming at precisely this lofty goal, under the rubric of automated programming or program synthesis (Gulwani et al. 2017, Smith 1984). Most relevant are algorithms for inductive program synthesis that attempt to automatically construct a program from examples of the program's desired execution (Gulwani et al. 2015). In contrast to the inefficiencies of blind, random search, these algorithms take a small step toward the child as hacker by employing smarter and far more efficient goal-directed search in the space of programs. They use strategies inspired loosely by the problem-solving techniques human programmers use, such as divide and conquer (Alur et al. 2017, Smith 1985), backward chaining of goal–subgoal constraints or constraints on types of functions and the inputs they require (Osera & Zdancewic 2015, Polozov & Gulwani 2015), and higher-order program templates to guide search and form abstractions (Cropper & Muggleton 2017, Lin et al. 2014, Solar-Lezama 2009). Recently, program induction has also looked to more implicit strategies for guiding search based on machine learning: discovering patterns in program outputs diagnostic of the program's internal structure for previously solved tasks (Balog et al. 2017, Devlin et al. 2017, Nye et al. 2019).

These smarter search approaches are just beginning to be applied to learning programs in a hierarchical Bayesian framework (Dechter et al. 2013; Ellis et al. 2016, 2018, 2020; Rule et al. 2018; Ullman et al. 2018) and learning probabilistic generative programs (Ellis et al. 2020, Hewitt et al. 2020), as would be needed to model intuitive theories. To date, these methods can synthesize only very simple, short programs—more like fragments of a theory, or additions to it, rather than a whole new theory synthesized from scratch. But these efforts are still in their infancy. Much more research is needed to build program induction systems with all the problem-solving strategies available to children, and to study what they can (and cannot) learn given sufficient data and time.

5. FROM INTUITIVE THEORIES TO SCIENTIFIC THEORIES

While we have focused on the ways children build models of the world, our framework is more general: It provides a way to think about any model-building agent in computational terms. In particular, it applies just as well in principle to the scientist as it does to the child as scientist. Both formal scientific theories and intuitive theories aim to solve the same problem: finding generative programs that best account for a body of data (cf. Li & Vitányi 2008). Formal and intuitive theories both operate on multiple levels of abstraction (Kuhn 2012, Wellman & Gelman 1998), and the halting, noisy dynamics of theory change across levels in both science (Kuhn 2012, Nersessian 1992) and development (Carey 2009, Siegler 2007) have been cast as the dynamics of computationally bounded, hierarchical Bayesian learning (Henderson et al. 2010, Ullman et al. 2012).

These analogies are to be expected, given that science was the inspiration for this view of development, and they might help explain why science is even possible for human beings—let alone so appealing for many of us, and so successful as a cultural innovation. But they also raise essential tensions. If children are little scientists (or if scientists are big children; Gopnik & Wellman 1992), we need to explain why science is so hard to do and so hard to teach. If children all over were building intuitive theories tens of thousands of years ago, we need to explain why capital-S Science is often traced to a particular moment in time, such as the formalization of experimentation as a method several hundred years ago, or the move from agents to objects as the basis for explanation (e.g., Thagard 2008).

These tensions might be eased by considering that the same cognitive processes of model building could have very different dynamics and outcomes in different circumstances. For the sake of plainer argument, suppose the mind holds a single program-learning mechanism (PLM). This PLM takes in observable data and returns a probabilistic generative program to account for the data. The PLM does not start every search from scratch but instead rationally reuses programs it has already learned. The output of this PLM will vary wildly depending on the primitives it starts with and the input it is tasked to explain. If the PLM inherits through evolution useful inductive biases in the form of relevant data representations, input analyzers, and helpful primitives—all selected for being relevant for human-scale, frequently encountered domains such as intuitive physics, psychology, or biology—then it will have an easier time discovering a relevant theory, and most learners will converge on the same programs because they start from the same place.

But if the very same PLM is directed at a domain outside the realm of everyday experience, where its inductive bias was not selected for and may not apply, then several differences will naturally unfold. First, the problem itself will be much, much harder. Search will be longer and progress more difficult. Second, any newly discovered useful variables and functions will need to be passed on pedagogically or culturally, not through biological inheritance. Third, as a community of learners initially explores a new domain, its problem will not be a lack of theories but an overabundance. Because the space of primitives is not shared, the PLMs of different learners may invent different subroutines, variables, and data formats. Disagreements will abound regarding proposed programs and even what the relevant input to these programs should be.

Still, it would be wrong to think of scientific theory building as pointing a blank-slate PLM willy-nilly at haphazard piles of data. The shared endowments of evolution, development, and culture shape both the search for new concepts and the understanding of concepts others have discovered. In searching for concepts, the PLM will make proposals by drawing on already established libraries, as new scientific proposals often rely on intuitive mental models and pictures ("suppose electrons are balls" or "suppose the magnetic field is a fluid with little vortices"; cf. Dirac 1963, Nersessian 1992). Even if such pictures do not directly correspond to reality, they may still be useful as initial scaffolding steps for developing a scientific theory. That new scientific concepts rely on commonly endowed implicit background knowledge is not a new proposal (Mach 1910), but the approaches developed here could also help us better understand the cognitive processes of early science in computational terms.

As for understanding new concepts that others have discovered, a PLM given an existing, vetted program may still try to reformulate it in terms of its existing libraries. Consider Ohm's law, which relates current, resistance, and voltage ($I = V/R$). It can be regarded as a short program relating three variables in a way that compresses and predicts a wide range of phenomena. There is no need to find a better or shorter program. And yet, for a student learning this law, merely coming to manipulate these symbols well enough to solve homework problems may not lead to understanding what they really mean. Now, imagine that we tell the student to think of current as the flow of a fluid (Esposito 1969)—resistors inhibit the flow, voltage is the pressure, and so on. This does not make the original miniprogram any shorter or more predictive, and an ideal PLM could reject it. And yet, the student may find it helpful and meaningful, because she already has intuitive physics programs that can simulate and explain human-scale fluid behavior (Bates et al. 2019). While some scientists may resist the urge to reinterpret a formal theory through the lens of existing intuitive concepts, the urge itself exists and is apparent in the way even scientists speak informally. We use our intuitive physics and psychology to say things like "photons bounce off the mirror" (photons aren't balls), "T cells don't want to harm healthy cells" (T cells don't want anything), or even "the electron wants to expand the sublattice" (electrons can't want anything).

In our view, scientists are both rightly skeptical of and reasonably drawn to such intuitively compelling metaphors—especially those that attribute intentional agency where none is evidently present. On the one hand, the story of science is often told as a sequence of hard-won moves from explanations based on agents toward accounts that explain the same phenomena mechanistically, with no need to posit any mysterious intentional force pulling levers behind the curtain. On the other hand, positing a hidden agent acting with some goal in mind can provide a rational (compressive, predictive, generalizable) explanation for phenomena when there is no compelling alternative. Infants and young children rationally posit hidden agents to explain the spontaneous creation of order out of disorder or the appearance of an object or structure where none was present before (Newman et al. 2010, Saxe et al. 2005). So do detectives and spies (Eco & Sebeok 1988, Fleming 1959). Even in science, in the context of paradigms that explicitly deny an intentional explanation for some class of physical, biological, or social phenomena, metaphors such as "the invisible hand" (Smith 1776) or "the blind watchmaker" (Dawkins 1986) can elegantly capture the workings of nonintentional processes, in ways that motivate the most explanatory versions of the mechanistic theories scientists ultimately arrive at.

Thus, the difficulties of science may not be so difficult to explain, as they are the natural difficulties of trying to solve the hard problems of program search that children solve, without all of the supports that children have available to them. Learning in development is reasonably fast, reliable, and robust only because we stand on the shoulders of two giants, biological and cultural evolution. And even when scientific theories do not ask for it, intuitive theories can't help trying to give science a leg up.

6. LOOKING BACK AND LOOKING AHEAD

Bayesian inference provides a general frame in which to think about how rational agents can build models of their world. The specific tools of hierarchical Bayesian inference over structured representations, and learning as a goal-directed search for probabilistic generative programs, give us a way to understand the processes of model building for agents with minds like humans—that is, with rich capacities for mental simulation and abstract thought but also severe constraints on time, energy, and computational resources. These tools can provide insight into the central questions of human cognitive development, what knowledge we as humans start with, and how we get the rest, as well as normative answers for how to build human-like learning machines.

Yet a full account of how humans come to their models of the world will have to grapple with a much bigger picture. Children and adults, engineers and scientists, cultures and societies, biological and artificial evolution all come up with models to explain experience and, in some sense, face the same computational challenge. It is no wonder the specifics vary greatly, as these processes unfold over different timescales, with different experiences, using different primitives, under different constraints. There is much to explore in future research across this space of model building and model builders.

Closer to home, we are still far from having a fully working computational model of cognitive development: a human-like program-learning program that could start with the program primitives infants do and, with the experiences of only a few years in this world, build all the models that children do. But we have our own scientific models to work with, and promising next steps. This is reason enough to hope that we are, at least, searching in the right space.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

We thank Fei Xu, Alison Gopnik, Henry Wellman, Charles Kemp, Amy Perfors, Noah Goodman, Vikash Mansinghka, and Steven Piantadosi, and especially Susan Carey, Elizabeth Spelke, Laura Schulz, and Josh Rule, for many discussions that helped shape our thinking in this article and well beyond. We are grateful for financial support from the Center for Brains, Minds and Machines through a National Science Foundation Science and Technology grant (award CCF-1231216), the Siegel Family Endowment, and the US Defense Advanced Research Projects Agency Machine Common Sense program.

LITERATURE CITED

Alur R, Radhakrishna A, Udupa A. 2017. Scaling enumerative program synthesis via divide and conquer. In *Proceedings of the 23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (*TACAS 2017*), pp. 319–36. Berlin: Springer

Anderson EM, Hespos SJ, Rips LJ. 2018. Five-month-old infants have expectations for the accumulation of nonsolid substances. *Cognition* 175:1–10

Anderson JR. 1990. *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum

Baillargeon R, Li J, Gertner Y, Wu D. 2011. How do infants reason about physical events? In *The Wiley-Blackwell Handbook of Childhood Cognitive Development*, ed. U Goswami, pp. 11–48. Oxford, UK: Blackwell. 2nd ed.

Baillargeon R, Li J, Ng W, Yuan S. 2008. An account of infants' physical reasoning. See Woodward & Needham 2008, pp. 66–116

Baillargeon R, Spelke ES, Wasserman S. 1985. Object permanence in five-month-old infants. *Cognition* 20:191–208

Baker CL, Jara-Ettinger J, Saxe R, Tenenbaum JB. 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nat. Hum. Behav.* 1:0064

Baker CL, Saxe R, Tenenbaum JB. 2009. Action understanding as inverse planning. *Cognition* 113:329–49

Balog M, Gaunt AL, Brockschmidt M, Nowozin S, Tarlow D. 2017. DeepCoder: learning to write programs. arXiv:1611.01989 [cs.LG]

Bates CJ, Yildirim I, Tenenbaum JB, Battaglia P. 2019. Modeling human intuitions about liquid flow with particle-based simulation. *PLOS Comput. Biol.* 15:e1007210

Battaglia PW, Hamrick JB, Tenenbaum JB. 2013. Simulation as an engine of physical scene understanding. *PNAS* 110:18327–32

Baum EB. 2004. *What Is Thought*? Cambridge, MA: MIT Press

Bonawitz E, Ullman TD, Bridgers S, Gopnik A, Tenenbaum JB. 2019. Sticking to the evidence? A behavioral and computational case study of micro-theory change in the domain of magnetism. *Cogn. Sci.* 43:e12765

Carey S. 1985. *Conceptual Change in Childhood*. Cambridge, MA: MIT Press/Bradford

Carey S. 2009. *The Origin of Concepts*. Oxford, UK: Oxford Univ. Press

Carey S, Spelke E. 1994. Domain-specific knowledge and conceptual change. In *Mapping the Mind: Domain Specificity in Cognition and Culture*, ed. LA Hirschfeld, SA Gelman, pp. 169–200. Cambridge, UK: Cambridge Univ. Press

Carey S, Spelke E. 1996. Science and core knowledge. *Philos. Sci.* 63:515–33

Chater N, Oaksford M. 2013. Programs as causal models: speculations on mental programs and mental representation. *Cogn. Sci.* 37:1171–91

Chu J, Schulz LE. 2020. Play, curiosity, and cognition. *Annu. Rev. Dev. Psychol.* 2:317–43

Craik K. 1943. *The Nature of Explanation*. Cambridge, UK: Cambridge Univ. Press

Cropper A, Muggleton SH. 2017. Learning higher-order logic programs through abstraction and invention. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 1418–24. Palo Alto, CA: AAAI

Csibra G. 2008. Goal attribution to inanimate agents by 6.5-month-old infants. *Cognition* 107:705–17

Csibra G, Biró S, Koós O, Gergely G. 2003. One-year-old infants use teleological representations of actions productively. *Cogn. Sci.* 27:111–33

Czégel D, Zachar I, Szathmáry E. 2018. Major evolutionary transitions as Bayesian structure learning. bioRxiv 359596. **https://doi.org/10.1101/359596**

Dawkins R. 1986. *The Blind Watchmaker*. London: Norton

Dechter E, Malmaud J, Adams RP, Tenenbaum JB. 2013. Bootstrap learning via modular concept discovery. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 1302–9. Menlo Park, CA: AAAI

Devlin J, Uesato J, Bhupatiraju S, Singh R, Mohamed AR, Kohli P. 2017. RobustFill: neural program learning under noisy I/O. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 990–98. New York: ACM

Dewar KM, Xu F. 2010. Induction, overhypothesis, and the origin of abstract knowledge: evidence from 9-month-old infants. *Psychol. Sci.* 21:1871–77

Dirac PAM. 1963. The evolution of the physicist's picture of nature. *Sci. Am.* 208:45–53

Eco U, Sebeok TA, eds. 1988. *The Sign of Three: Dupin, Holmes, Peirce*. Bloomington: Indiana Univ. Press

Ellis K, Morales L, Sablé-Meyer M, Solar-Lezama A, Tenenbaum J. 2018. Learning libraries of subroutines for neurally-guided Bayesian program induction. In *Advances in Neural Information Processing Systems 31*, ed. S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett, pp. 7805–15. Red Hook, NY: Curran

Ellis K, Solar-Lezama A, Tenenbaum J. 2016. Sampling for Bayesian program learning. In *Advances in Neural Information Processing Systems 29*, ed. DD Lee, M Sugiyama, UV Luxburg, I Guyon, R Garnett, pp. 1297–305. Red Hook, NY: Curran

Ellis K, Wong C, Nye M, Sablé-Meyer M, Cary L, et al. 2020. DreamCoder: growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. arXiv:2006.08381 [cs.AI]

Erdogan G, Jacobs RA. 2017. Visual shape perception as Bayesian inference of 3D object-centered shape representations. *Psychol. Rev.* 124:740–61

Esposito A. 1969. A simplified method for analyzing hydraulic circuits by analogy. *Mach. Des.* 41:173–77

Fleming I. 1959. *Goldfinger*. London: Cape

Fodor JA. 1975. *The Language of Thought*. Cambridge, MA: Harvard Univ. Press

Fodor JA. 1998. *Concepts: Where Cognitive Science Went Wrong*. New York: Oxford Univ. Press

Forbus K. 2019. *Qualitative Representations: How People Reason and Learn About the Continuous World*. Cambridge, MA: MIT Press

Gelman SA. 2009. Learning from others: children's construction of concepts. *Annu. Rev. Psychol.* 60:115–40

Gelman SA, Carlin JB, Stern HS, Rubin DB. 2013. *Bayesian Data Analysis*. New York: Chapman & Hall. 3rd ed.

Geman S, Geman D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6:721–41

Gentner D. 1989. The mechanisms of analogical learning. In *Similarity and Analogical Reasoning*, ed. S Vosniadou, A Ortony, pp. 199–241. Cambridge, UK: Cambridge Univ. Press

Gentner D, Stevens AL, eds. 1983. *Mental Models*. Hillsdale, NJ: Erlbaum

Gershman SJ, Goodman ND. 2014. Amortized inference in probabilistic reasoning. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pp. 517–22. Austin, TX: Cogn. Sci. Soc.

Gershman SJ, Horvitz EJ, Tenenbaum JB. 2015. Computational rationality: a converging paradigm for intelligence in brains, minds, and machines. *Science* 349:273–78

Gerstenberg T, Tenenbaum JB. 2017. Intuitive theories. In *Oxford Handbook of Causal Reasoning*, ed. MR Waldmann, pp. 515–48. Oxford, UK: Oxford Univ. Press

Gilden DL, Proffitt DR. 1989. Understanding collision dynamics. *J. Exp. Psychol. Hum. Percept. Perform.* 15:372–83

Goodman ND. 1983. *Fact, Fiction, and Forecast*. Cambridge, MA: Harvard Univ. Press

Goodman ND, Mansinghka VK, Roy DM, Bonawitz K, Tenenbaum JB. 2008. Church: a language for generative models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pp. 220–29. New York: ACM

Goodman ND, Tenenbaum JB, Gerstenberg T. 2014. *Concepts in a probabilistic language of thought*. Tech. Rep., Cent. Brains Minds Mach., MIT, Cambridge, MA

Goodman ND, Tenenbaum JB, ProbMods Contrib. 2016. *Probabilistic Models of Cognition*. **http://probmods.org**. 2nd ed.

Goodman ND, Ullman T, Tenenbaum J. 2011. Learning a theory of causality. *Psychol. Rev.* 118:110–19

Gopnik A. 1996. The scientist as child. *Philos. Sci.* 63:485–514

Gopnik A, Glymour C, Sobel D, Schulz L, Kushnir T, Danks D. 2004. A theory of causal learning in children: causal maps and Bayes nets. *Psychol. Rev.* 111:1–31

Gopnik A, Griffiths TL, Lucas CG. 2015. When younger learners can be better (or at least more open-minded) than older ones. *Curr. Dir. Psychol. Sci.* 24:87–92

Gopnik A, Meltzoff AN. 1997. *Words, Thoughts, and Theories*. Cambridge, MA: MIT Press

Gopnik A, O'Grady S, Lucas CG, Griffiths TL, Wente A, et al. 2017. Changes in cognitive flexibility and hypothesis search across human life history from childhood to adolescence to adulthood. *PNAS* 114:7892–99

Gopnik A, Wellman HM. 1992. Why the child's theory of mind really is a theory. *Mind Lang.* 7:145–71

Gopnik A, Wellman HM. 1994. The theory theory. In *Mapping the Mind: Domain Specificity in Cognition and Culture*, ed. LA Hirschfeld, SA Gelman, pp. 257–93. Cambridge, UK: Cambridge Univ. Press

Gopnik A, Wellman HM. 2012. Reconstructing constructivism: causal models, Bayesian learning mechanisms, and the theory theory. *Psychol. Bull.* 138:1085–108

Gregory J. 2018. *Game Engine Architecture*. Boca Raton, FL: CRC

Griffiths TL, Chater N, Kemp C, Perfors A, Tenenbaum JB. 2010. Probabilistic models of cognition: exploring representations and inductive biases. *Trends Cogn. Sci.* 14:357–64

Griffiths TL, Kemp C, Tenenbaum JB. 2008. Bayesian models of cognition. In *The Cambridge Handbook of Computational Psychology*, ed. R Sun, pp. 59–100. Cambridge, UK: Cambridge Univ. Press

Griffiths TL, Lieder F, Goodman ND. 2015. Rational use of cognitive resources: levels of analysis between the computational and the algorithmic. *Top. Cogn. Sci.* 7:217–29

Griffiths TL, Tenenbaum JB. 2005. Structure and strength in causal induction. *Cogn. Psychol.* 51:285–386

Griffiths TL, Tenenbaum JB. 2009. Theory-based causal induction. *Psychol. Rev.* 116:661–716

Griffiths TL, Vul E, Sanborn AN. 2012. Bridging levels of analysis for probabilistic models of cognition. *Curr. Dir. Psychol. Sci.* 21:263–68

Gulwani S, Hernndez-Orallo J, Kitzelmann E, Muggleton HS, Schmid U, Zorn GB. 2015. Inductive programming meets the real world. *Commun. ACM* 58(11):90–99

Gulwani S, Polozov A, Singh R. 2017. *Program Synthesis: Foundations and Trends in Programming Languages*, Vol. 4. Delft, Neth.: Now

Hamrick JB, Battaglia PW, Griffiths TL, Tenenbaum JB. 2016. Inferring mass in complex scenes by mental simulation. *Cognition* 157:61–76

Henderson L, Goodman ND, Tenenbaum JB, Woodward JF. 2010. The structure and dynamics of scientific theories: a hierarchical Bayesian perspective. *Philos. Sci.* 77:172–200

Hewitt LB, Le TA, Tenenbaum JB. 2020. *Learning to learn generative models with Memoised Wake-Sleep*. Paper presented at the Conference on Uncertainty in Artificial Intelligence, online, Aug. 3–6

Jara-Ettinger J, Gweon H, Schulz LE, Tenenbaum JB. 2016. The nave utility calculus: computational principles underlying commonsense psychology. *Trends Cogn. Sci.* 20:589–604

Jaynes ET. 2003. *Probability Theory: The Logic of Science*. Cambridge, UK: Cambridge Univ. Press

Johnson-Laird PN. 2004. The history of mental models. In *Psychology of Reasoning*, ed. K Manktelow, MC Chung, pp. 189–222. New York: Psychology

Johnston AM, Sheskin M, Keil FC. 2019. Learning the relevance of relevance and the trouble with truth: evaluating explanatory relevance across childhood. *J. Cogn. Dev.* 20:555–72

Kemp C. 2008. *The acquisition of inductive constraints*. PhD Thesis, MIT, Cambridge, MA

Kemp C, Goodman ND, Tenenbaum JB. 2008. Theory acquisition and the language of thought. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pp. 1606–11. Austin, TX: Cogn. Sci. Soc.

Kemp C, Goodman ND, Tenenbaum JB. 2010. Learning to learn causal models. *Cogn. Sci.* 34:1185–243

Kemp C, Perfors A, Tenenbaum J. 2007. Learning overhypotheses with hierarchical Bayesian models. *Dev. Sci.* 10:307–21

Kemp C, Tenenbaum JB. 2008. The discovery of structural form. *PNAS* 105:10687–92

Kemp C, Tenenbaum JB. 2009. Structured statistical models of inductive reasoning. *Psychol. Rev.* 116:20–58

Kiley Hamlin J, Ullman T, Tenenbaum JB, Goodman N, Baker C. 2013. The mentalistic basis of core social cognition: experiments in preverbal infants and a computational model. *Dev. Sci.* 16:209–26

Kirkpatrick S, Gelatt CD, Vecchi MP. 1983. Optimization by simulated annealing. *Science* 220:671–80

Koza JR. 1994. *Genetic Programming II*. Cambridge, MA: MIT Press

Kubricht J, Zhu Y, Jiang C, Terzopoulos D, Zhu S, Lu H. 2017. Consistent probabilistic simulation underlying human judgment in substance dynamics. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, pp. 700–5. Austin, TX: Cogn. Sci. Soc.

Kuhn TS. 2012. *The Structure of Scientific Revolutions*. Chicago: Univ. Chicago Press

Lake BM, Salakhutdinov R, Tenenbaum JB. 2015. Human-level concept learning through probabilistic program induction. *Science* 350:1332–38

Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. 2017. Building machines that learn and think like people. *Behav. Brain Sci.* 40:e253

Landau B, Gleitman LR. 1985. *Language and Experience: Evidence from the Blind Child*. Cambridge, MA: Harvard Univ. Press

Lerer A, Gross S, Fergus R. 2016. Learning physical intuition of block towers by example. arXiv:1603.01312 [cs.AI]

Levin LA. 1973. Universal sequential search problems. *Probl. Pereda. Inf.* 9:115–16

Li M, Vitányi P. 2008. *An Introduction to Kolmogorov Complexity and Its Applications*. Berlin: Springer. 3rd ed.

Lieder F, Griffiths TL. 2020. Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behav. Brain Sci.* 43:e1

Lin D, Dechter E, Ellis K, Tenenbaum JB, Muggleton S. 2014. Bias reformulation for one-shot function induction. In *Proceedings of the 21st European Conference on Artificial Intelligence*, pp. 525–30. Amsterdam: IOS

Liu S, Ullman TD, Tenenbaum JB, Spelke ES. 2017. Ten-month-old infants infer the value of goals from the costs of actions. *Science* 358:1038–41

Lombrozo T. 2016. Explanatory preferences shape learning and inference. *Trends Cogn. Sci.* 20:748–59

Lucas CG, Griffiths TL, Xu F, Fawcett C, Gopnik A, et al. 2014. The child as econometrician: a rational model of preference understanding in children. *PLOS ONE* 9:e92160

Mach E. 1910. *Populär-wissenschaftliche Vorlesungen*. Leipzig, Ger.: Barth

MacKay DJ. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge Univ. Press

Marr D. 1982. *Vision*. London: Freeman

Murphy GL, Medin DL. 1985. The role of theories in conceptual coherence. *Psychol. Rev.* 92:289–316

Nersessian NJ. 1992. How do scientists think? Capturing the dynamics of conceptual change in science. *Cogn. Models Sci.* 15:3–44

Newman GE, Keil FC, Kuhlmeier VA, Wynn K. 2010. Early understandings of the link between agents and order. *PNAS* 107:17140–45

Nye M, Hewitt L, Tenenbaum J, Solar-Lezama A. 2019. Learning to infer program sketches. *J. Mach. Learn. Res.* 97:4861–70

Osera PM, Zdancewic S. 2015. Type-and-example-directed program synthesis. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 619–30. New York: ACM

Pearl J. 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Berlin: Springer

Pearl J. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge, UK: Cambridge Univ. Press

Perfors A, Tenenbaum JB, Griffiths TL, Xu F. 2011. A tutorial introduction to Bayesian models of cognitive development. *Cognition* 120:302–21

Perfors A, Tenenbaum JB, Wonnacott E. 2010. Variability, negative evidence, and the acquisition of verb argument constructions. *J. Child Lang.* 37:607–42

Phillips J, Morris A, Cushman F. 2019. How we know what not to think. *Trends Cogn. Sci.* 23:1026–40

Piantadosi ST, Tenenbaum JB, Goodman ND. 2012. Bootstrapping in a language of thought: a formal model of numerical concept learning. *Cognition* 123:199–217

Piantadosi ST, Tenenbaum JB, Goodman ND. 2016. The logical primitives of thought: empirical foundations for compositional cognitive models. *Psychol. Rev.* 123:392–424

Polozov O, Gulwani S. 2015. FlashMeta: a framework for inductive program synthesis. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications* (*OOPSLA 2015*), pp. 107–26. New York: ACM

Rule JS, Schulz E, Piantadosi ST, Tenenbaum JB. 2018. Learning list concepts through program induction. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pp. 2370–75. Austin, TX: Cogn. Sci. Soc.

Rule JS, Tenenbaum JB, Piantadosi ST. 2020. The child as hacker. *Trends Cogn. Sci.* In press

Russell SJ, Norvig P. 2002. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall. 2nd ed.

Russell SJ, Norvig P. 2020. *Artificial Intelligence: A Modern Approach*. London: Pearson. 4th ed.

Saad FA, Cusumano-Towner MF, Schaechtle U, Rinard MC, Mansinghka VK. 2019. Bayesian synthesis of probabilistic programs for automatic data modeling. *Proc. ACM Program. Lang.* 3:37

Sanborn AN, Mansinghka VK, Griffiths TL. 2013. Reconciling intuitive physics and Newtonian mechanics for colliding objects. *Psychol. Rev.* 120:411–37

Saxe R, Tenenbaum J, Carey S. 2005. Secret agents: inferences about hidden causes by 10- and 12-month-old infants. *Psychol. Sci.* 16:995–1001

Schulz L. 2012a. Finding new facts; thinking new thoughts. *Adv. Child Dev. Behav.* 43:269–94

Schulz L. 2012b. The origins of inquiry: inductive inference and exploration in early childhood. *Trends Cogn. Sci.* 16:382–89

Shi L, Griffiths TL, Feldman NH, Sanborn AN. 2010. Exemplar models as a mechanism for performing Bayesian inference. *Psychon. Bull. Rev.* 17:443–64

Siegler RS. 2007. Cognitive variability. *Dev. Sci.* 10:104–9

Siegler RS, Chen Z. 1998. Developmental differences in rule learning: a microgenetic analysis. *Cogn. Psychol.* 36:273–310

Simon HA. 1956. Rational choice and the structure of the environment. *Psychol. Rev.* 63:129–38

Simon HA. 1962. An information processing theory of intellectual development. *Monogr. Soc. Res. Child Dev.* 27:154–55

Smith A. 1776. *The Wealth of Nations*. London: Strahan

Smith DR. 1984. Synthesis of LISP programs from examples: a survey. In *Automatic Program Construction Techniques*, ed. AW Biermann, G Guiho, Y Kodratoff, pp. 307–24. London: Macmillan

Smith DR. 1985. Top-down synthesis of divide-and-conquer algorithms. *Artif. Intell.* 27:43–96

Smith KA, Mei L, Yao S, Wu J, Spelke E, et al. 2019. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. In *Advances in Neural Information Processing Systems 32*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett, pp. 8985–95. Red Hook, NY: Curran

Smith KA, Mei L, Yao S, Wu J, Spelke E, et al. 2020. The fine structure of surprise in intuitive physics: When, why, and how much? In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*, pp. 3048–54. Austin, TX: Cogn. Sci. Soc.

Solar-Lezama A. 2009. The sketching approach to program synthesis. In *Proceedings of the Asian Symposium on Programming Languages and Systems* (*ASPLAS*), pp. 4–13. Berlin: Springer

Spelke ES. 1990. Principles of object perception. *Cogn. Sci.* 14:29–56

Spelke ES. 1995. Initial knowledge: six suggestions. *Cognition* 50:433–47

Spelke ES, Breinlinger K, Macomber J, Jacobson K. 1992. Origins of knowledge. *Psychol. Rev.* 99:605–32

Spelke ES, Kestenbaum R, Simons DJ, Wein D. 1995. Spatiotemporal continuity, smoothness of motion and object identity in infancy. *Br. J. Dev. Psychol.* 13:113–42

Spelke ES, Kinzler K. 2007. Core knowledge. *Dev. Sci.* 10:89–96

Stahl AE, Feigenson L. 2015. Observing the unexpected enhances infants' learning and exploration. *Science* 348:91–94

Stanley KO, Clune J, Lehman J, Miikkulainen R. 2019. Designing neural networks through neuroevolution. *Nat. Mach. Intell.* 1:24–35

Sutton RS, Barto AG. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press

Téglás E, Vul E, Girotto V, Gonzalez M, Tenenbaum JB, Bonatti LL. 2011. Pure reasoning in 12-month-old infants as probabilistic inference. *Science* 332:1054–59

Tenenbaum JB, Griffiths TL, Kemp C. 2006. Theory-based Bayesian models of inductive learning and reasoning. *Trends Cogn. Sci.* 10:309–18

Tenenbaum JB, Griffiths TL, Niyogi S. 2007. Intuitive theories as grammars for causal inference. In *Causal Learning: Psychology, Philosophy, and Computation*, ed. A Gopnik, L Schulz, pp. 301–22. Oxford, UK: Oxford Univ. Press

Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND. 2011. How to grow a mind: statistics, structure, and abstraction. *Science* 331:1279–85

Thagard P. 2008. Conceptual change in the history of science: life, mind, and disease. In *International Handbook of Research on Conceptual Change*, ed. S Vosniadou, pp. 374–87. New York: Taylor & Francis

Turing AM. 2009. Computing machinery and intelligence. In *Parsing the Turing Test*, ed. R Epstein, G Roberts, G Beber, pp. 23–65. Berlin: Springer

Ullman TD. 2015. *On the nature and origin of intuitive theories: learning, physics and psychology*. PhD Thesis, MIT, Cambridge, MA

Ullman TD, Goodman ND, Tenenbaum JB. 2012. Theory learning as stochastic search in the language of thought. *Cogn. Dev.* 27:455–80

Ullman TD, Siegel MH, Tenenbaum JB, Gershman S. 2016. Coalescing the vapors of human experience into a viable and meaningful comprehension. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, pp. 1493–98. Austin, TX: Cogn. Sci. Soc.

Ullman TD, Spelke E, Battaglia P, Tenenbaum JB. 2017. Mind games: game engines as an architecture for intuitive physics. *Trends Cogn. Sci.* 21:649–65

Ullman TD, Stuhlmüller A, Goodman ND, Tenenbaum JB. 2018. Learning physical parameters from dynamic scenes. *Cogn. Psychol.* 104:57–82

Vul E, Goodman N, Griffiths TL, Tenenbaum JB. 2014. One and done? Optimal decisions from very few samples. *Cogn. Sci.* 38:599–637

Waxman SR, Markow DB. 1995. Words as invitations to form categories: evidence from 12- to 13-month-old infants. *Cogn. Psychol.* 29:257–302

Wellman HM. 2014. *Making Minds: How Theory of Mind Develops*. Oxford, UK: Oxford Univ. Press

Wellman HM, Fang F, Peterson CC. 2011. Sequential progressions in a theory-of-mind scale: longitudinal perspectives. *Child Dev*. 82:780–92

Wellman HM, Gelman S. 1998. Knowledge acquisition in foundational domains. In *Handbook of Child Psychology*, Vol. 2: *Cognition*, *Perception*, *and Language*, ed. W Damon, pp. 523–73. New York: Wiley. 6th ed.

Woodward AL. 1998. Infants selectively encode the goal object of an actor's reach. *Cognition* 69:1–34

Woodward AL. 1999. Infants' ability to distinguish between purposeful and non-purposeful behaviors. *Infant Behav. Dev.* 22:145–60

Woodward AL, Needham A, eds. 2008. *Learning and the Infant Mind*. Oxford, UK: Oxford Univ. Press

Wu J, Yildirim I, Lim JJ, Freeman B, Tenenbaum J. 2015. Galileo: perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems 28*, ed. C Cortes, ND Lawrence, DD Lee, M Sugiyama, R Garnett, pp. 127–35. Red Hook, NY: Curran

Xu F. 2019. Towards a rational constructivist theory of cognitive development. *Psychol. Rev.* 126:841–64

Xu F, Tenenbaum JB. 2007. Word learning as Bayesian inference. *Psychol. Rev.* 114:245–62

Annual Review of Developmental Psychology

Volume 2, 2020

Contents

Errata

An online log of corrections to *Annual Review of Developmental Psychology* articles may be found at http://www.annualreviews.org/errata/devpsych