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

Tomer D. Ullman[a,c], Joshua B. Tenenbaum[b,c]

[a]Department of Psychology, Harvard University, Cambridge, Massachusetts, 02138; tullman@fas.harvard.edu
[b]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139; jbt@mit.edu
[c]Center for Brains, Minds, and Machines (CBMM)

Abstract

A Bayesian framework helps to 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: (i) Core knowledge in infancy, and (ii) 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 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.

Keywords: cognitive development, bayesian models, intuitive theories, program learning, computational modeling, child as hacker, core knowledge

Preprint of paper submitted to Annual Review of Developmental Psychology

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 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 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 taken recently towards answering these questions 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's happened several times before.

We will focus in particular on the toolkit of hierarchical Bayesian models defined over structured representations, especially probabilistic generative programs which provide compelling ways to express, learn, and reason about knowledge that is abstract, causal, and generalizable. We structure our review in four parts, addressing four specific ways these tools can help to answer the big questions. First, 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 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 two examines what initial knowledge is needed to get model-building off the ground in infancy, and 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 three 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 times scales under different constraints than the evolutionary process that led to core knowledge. We also consider the algorithmic challenge of searching through vast spaces of candidate theories, and mechanisms children might use to navigate these spaces: learning as stochastic search, and learning as programming (or the "child as hacker"). Finally, we move from the child-as-scientist to the scientist-as-scientist, and how the Bayesian approach helps us understand both the deep similarities and fundamental tensions between these two modes of model-building: If cognitive development is like science, why is science often so hard, unreliable, and counter-intuitive, while the growth of commonsense thought feels – at least in retrospect – effortless and inevitable? We close with a more general perspective on the activity of model-building, and the lacunae and limitations in our own attempt to build models of it, which we hope to see addressed in future work.

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' 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.

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 (Fig. 1). Each point on it 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 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' Rule describes how we *should* move the mountains and valleys in light of observations. For a given model m, and evidence e, we have:

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

The posterior probability $P(m|e)$ is our degree of belief in model m conditioned on observing evidence e. This is proportional to the likelihood $P(e|m)$, or probability of observing the evidence 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 just the average of the likelihoods $P(e|m')$ from all other models m' in the hypothesis space M weighted by their priors

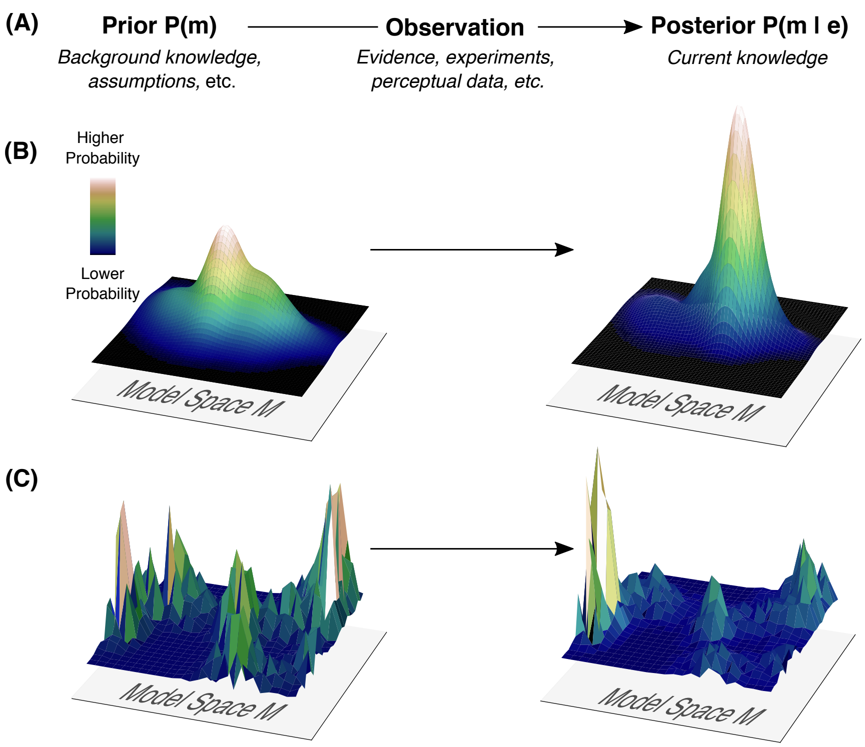


Figure 1: **(A)** The logic of Bayesian updating. The prior probability over possible models $P(m)$ shifts when new evidence e is observed, and forms a posterior probability distribution $P(m|e)$. **(B)** Schematic of smooth space of models representing relatively simple parameter change. **(C)** Schematic of discrete space of models, corresponding to harder to search through conceptual spaces.

$P(m')$. The posterior can thus also be written simply as the product of the prior times a term $R(e|m)$ measuring relative likelihood: To the extent that model m explains the evidence better (or worse) than the average model, $R(e|m)$ will be greater or less than 1, thus shifting the posterior probability $P(m|e)$ higher or lower relative to the prior $P(m)$ in response to what we observe. These are just the basics, presented informally. More thorough, formal treatments can found in Jaynes (2003); Gelman et al. (2013); Russell and Norvig (2020).

2.1. Hierarchical Bayesian models and domain knowledge

Hierarchical Bayesian models (HBMs) extend this picture to a nested tower of inferences: hypothesis spaces of hypothesis spaces, and priors on priors. These over-hypotheses and hyper-priors capture general beliefs that apply 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. Hierarchical models thus are 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., 2011b). 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 give a general formalism for what must be built in (the most abstract over-hypotheses and primitives) and what can be learned (potentially, all lower-level constraints and hypotheses).

We next outline a simple example illustrating these points (Fig. 2, left), but for more detail and further applications of HBMs in cognition and development we recommend readers to Kemp et al. (2007); Kemp (2008); Goodman et al. (2011b); Gopnik and Wellman (2012); Griffiths et al. (2008); Tenenbaum et al. (2006, 2011); Dewar and Xu (2010), 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; Kemp et al., 2007). Before drawing 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 wouldn't 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 $P(\texttt{next marble} = \texttt{green} | \texttt{6 green marbles}, \texttt{background assumptions})$ is relatively high, and much higher than the prior $P(\texttt{next marble} = \texttt{green})$ 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 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 drew a single marble from the fourth bag and found that it was 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 fall out naturally from a hierarchical Bayesian analysis. Your specific guess for the color of the marbles in a bag is informed by marbles drawn from that bag (evidence informs hypotheses). But assuming 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 (over-hypotheses shape hypotheses). This general understanding was formed from the same data

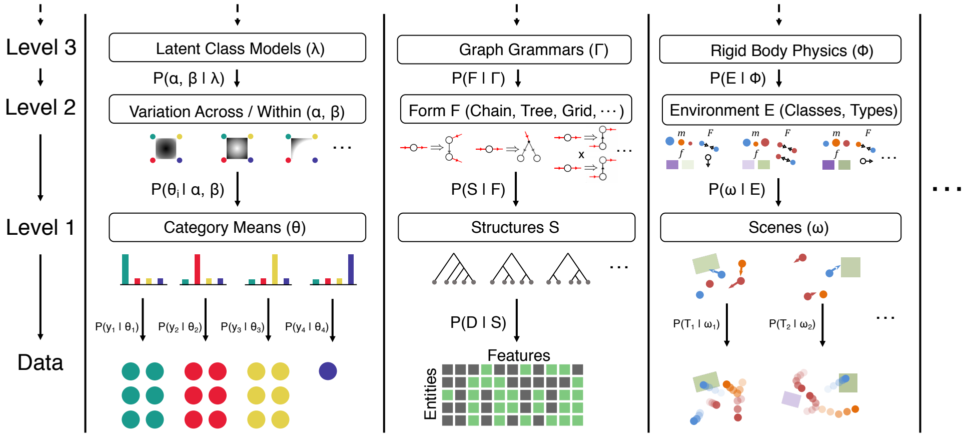


Figure 2: Examples of hierarchical generative models. Each level inherits from level above, which defines the distribution $P(Level_i|Level_{i+1})$. Higher levels capture more abstract, domain-general knowledge. **Left:** latent class models for discrete properties of objects (from Kemp et al., 2007), which can learn the distribution of a property within a category, while acquiring a general expectation about the distribution of properties. **Center:** graph grammars over structural forms (from Kemp and Tenenbaum, 2008), which can simultaneously learn e.g. that a tree model is more suited for describing a domain than a grid, and the specific tree models. **Right:** Rigid physics programs for dynamics scenes (from Ullman et al., 2018), which can reason about objects and dynamics.

you used to learn about the contents of specific bags (over-hypotheses and hypotheses can be learned at the same time). And you were reasonably justified in guessing an entire bag was filled with purple marbles from just a single purple observation, based on your understanding of the domain (over-hypotheses 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 (over-hypotheses 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-Multinomal 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 and Xu (2010) showed in a series of elegant studies on overhypothesis learning in nine-month-old infants.

Learning over-hypotheses on mixtures of latent properties is perhaps the simplest setting where HBMs have been applied, but more complex hierarchical models can be defined over richer and more diverse representational structures

(Tenenbaum et al., 2011; Griffiths et al., 2010). HBMs can even be used to infer the form of structure most appropriate for reasoning in a domain (Fig. 2, center). For example, tree-structured representations may be particularly useful for reasoning about the names and properties of object kinds (e.g. Xu and Tenenbaum, 2007; Kemp and Tenenbaum, 2009), 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 and Tenenbaum (2008) showed how these and other forms of structured probabilistic models – including cliques, chains, grids, rings, and many asymmetric directed models as well – can all be generated by graph grammars, with a meta-grammar 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. A hierarchical Bayesian learner can also infer that the entities in a domain could be organized differently for different purposes (Kemp and Tenenbaum, 2009). For example, animals could be organized in a tree structure to explain their anatomical and physiological properties, a dimensional structure to explain their behavioral and ecological properties, or a causal graph (e.g., a food web) to explain population properties and disease transmission

These examples based on mixtures, graphs, and grammars comprise only a few points in the landscape of possible ways to represent the over-hypotheses that people learn about domains. HBMs have also been defined on vector spaces, relational schemas, first-order and higher-order logic, to capture different aspects of domain structures. But most generally, all of these can be seen 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`, Goodman et al., 2016) provides a comprehensive introduction to probabilistic programs and their use in cognitive modeling, along with many interactive examples.

Besides unifying all of the familiar HBMs and over-hypothesis models discussed in above, probabilistic programs have given Bayesian computational accounts their first viable means to address 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. As these concepts have been the focus of much research in cognitive development, including both infants' core knowledge and children's developing intuitive theories, we focus on probabilistic programs in this setting.

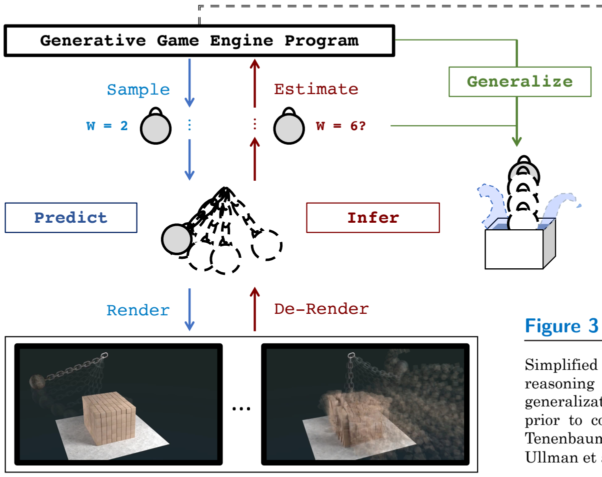
Informally, we can think of a probabilistic program as just a computer program that makes probabilistic choices. For a given input there is no single deterministic output, rather there is a probability distribution over outputs. We focus on *probabilistic generative programs* in which the program describes 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 backwards": observing some evidence that is the partial output of the program, and 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 notion that the mind builds internal models that mimic the world's causal structure, that 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 (see Craik (1943), but also Johnson-Laird (2004) for a more expansive history), and has shaped it since early days (Gentner and Stevens, 1983). Probabilistic generative programs can be seen as a modern incarnation of these ideas, which combines abstract, structured causal knowledge with capacities for Bayesian inference and simulation-based reasoning over those models (Gerstenberg and Tenenbaum, 2017). The approach has been applied across cognitive science in recent years, and especially in concept learning (Lake et al., 2015), causal and counterfactual reasoning (Chater and Oaksford, 2013; Gerstenberg and Tenenbaum, 2017), object perception (Erdogan and Jacobs, 2017; Wu et al., 2015), action understanding or intuitive psychology (Baker et al., 2017; Jara-Ettinger et al., 2016), and intuitive physics. We go into more detail on 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 re-imagine 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; Ullman et al., 2017; Sanborn et al., 2013), [1]

A generative program starts from positing the world that gives rise to percep-

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



```
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 shows part of program pseudo-code prior to conditioning (based on Battaglia, Hamrick, and Tenenbaum, 2013; Gerstenberg and Tenenbaum, 2017; Ullman et al., 2018).

tion. For intuitive physics, think of something like the physics-engine programs used in modern video games to create real-time interactive environments (see 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. An example is illustrated for one set of scenes in Fig. 3; see Fig. 2 (right) for a different physical domain framed as a 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 (Fig. 3), including: (i) *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 questions, such 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?' (ii) *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 (Fig. 2, right), from simple parameter estimation (how heavy is a block or the ball) to structural relations (the ball appears attached to the chain), to higher-level concept discovery (an unknown force appears to hold the blocks together). (iii) *Generalization.* Knowledge gained from one observed setting carries over to others. Having learned that the ball is heavy from collisions, we can modify parts of the program to predict how it will behave in a new setting, such as dropping it 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 Fig. 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 the cost of the full physics simulator, once enough data is 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 Eq. 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 et al., 2008) in a language that species models – 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 then just 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, and 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 are 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's brain is something like a notebook as one buys it from the stationers, rather little mechanism, and lots of blank sheets' (Turing, 2009). In other words, start without specific program content, only 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 discovers 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 over-correct, 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 pre-load a centipede with it, rather than having to learn to become a centipede each generation anew. But this is not the way to build a machine that is intelligent in any human-like fashion.

In between these extremes are many possibilities, but most promising, we believe, is a *Minimal Nativism.* This could also be what Turing had in mind when he hypothesized that 'One 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 than that for evolution, as these creatures can iteratively build and refine their models based on 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, content-less 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. Spelke and Kinzler, 2007; Spelke, 1990, 1995; Carey and Spelke, 1994; Spelke et al., 1992). The term "Core Knowledge" refers to early-emerging, possibly innate expectations 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 and 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 each time anew for cars, then jars, then guitars. Or take 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* (see e.g. Woodward, 1998, 1999; Csibra et al., 2003; Csibra, 2008; Liu et al., 2017).

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, and 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. To illustrate, 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 needed (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 is updated according to rules of dynamics, such as Newton's law $a = \frac{F}{m}$ 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, 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 fall out of the dynamics in particle-based simulations, which have been used to model people's intuitions about non-solid 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 judgements about how surprising a scene is, when the surprise occurs, and what kind of violation likely occurred (Smith et al., 2020). And a similar coarse probabilistic model predicts the looking times of 12-month-olds for a range of multi-object dynamic displays, varying in how much they violate continuity (Téglás et al., 2011).

Learning within core generative programs is limited, but possible, accounting for some of the learning that happens over the first year of life. Take again the program in Fig. 3. Such a program could start out uncertain over the weight of an object, and then after observing a collision 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 (see e.g. Ullman et al., 2018). This would correspond to allowing inference at higher levels of the HBM shown in Fig. 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 (Baillargeon et al., 2009, 2011, see e.g.), would then 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 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)

And finally, a full account of core knowledge as a start-up library of programs will need to explain how such knowledge could be discovered and encoded by evolution. Genetic programming methods are a promising computational direction (see e.g. Koza et al., 1994; Czégel et al., 2018; 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 constricts of core systems, children and adults can also 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 (see e.g. Carey, 1985; Murphy and Medin, 1985; Gopnik and Wellman, 1994; Gopnik and Meltzoff, 1997; Wellman and Gelman, 1998; Carey, 2009; Schulz, 2012b; Lombrozo, 2016). 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 subject to revision given new evidence. They are evaluated based on 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 2000's, the tools of hierarchical Bayesian models and then probabilistic programs have been extensively developed to capture the structure of intuitive theories and how they can be learned (Tenenbaum et al., 2011; Gopnik and Wellman, 2012; Goodman et al., 2014; Gerstenberg and Tenenbaum, 2017). These efforts build on but go importantly beyond earlier models of causal learning in children and adults (Gopnik et al., 2004; Griffiths and 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 (Tenenbaum et al., 2007; Kemp et al., 2010; Goodman et al., 2011a) – and 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 (Griffiths et al., 2010; Goodman et al., 2014).

4.1.1. The relationship between intuitive theories and core knowledge

The lines between core knowledge and intuitive theories, between learning in infancy and later in childhood, have been subject to intense debate. Some 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 and Spelke, 1996; Carey, 2009). 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 proto-conceptual primitives (Xu, 2019). In contrast, others have argued for continuity, or 'theories all the way down' (e.g. Gopnik, 1996; Gopnik and Wellman, 2012; Woodward and 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 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 means 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 within a mental physics engine, or new over-hypotheses over utility functions in a mental planning engine. And 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, and 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 (Kemp et al., 2008; Griffiths and Tenenbaum, 2009; 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 (Marr, 1982; Tenenbaum et al., 2011; Anderson, 1990). 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 pre-possessing a theory of mechanics before sitting under the 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 Fig. 1C, 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 head. Empirically, while different children eventually tend to converge on the same intuitive theories (see e.g. Carey, 2009; Gopnik and 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 and Chen, 1998; Siegler, 2007).

The ideal picture meets these challenges by moving to the algorithmic level of analysis (Marr, 1982; Griffiths et al., 2015), 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 (Griffiths et al., 2015; Gershman et al., 2015; Lieder and 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, just 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. Just because the combination of programs and program transformations theoretically express an infinite space does not mean the child does not learn or discover something new, in the same way that a child's ability to express and understand English doesn't mean that Shakespeare did not 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 and Norvig, 2020) has been frequently mined as inspiration for algorithmic-level Bayesian models of cognition (Griffiths et al., 2008; Goodman et al., 2008; Griffiths et al., 2012; Gershman et al., 2015), and is especially natural for learning probabilistic generative programs in a hierarchical Bayesian framework (Ullman et al., 2012, 2018; Saad et al., 2019).

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 distributions 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 (Piantadosi et al., 2012; Ullman et al., 2012; Bonawitz et al., 2019).

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 (Kirkpatrick et al., 1983; Geman and Geman, 1984), starting in a "hot" regime in which unlikely proposals (theories) that may be worse than the current 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 sub-optimal – but as the search cools, learners become less likely to change their theories, and especially to make large or sub-optimal changes. Thinking becomes more predictable, and if the search cools slowly enough, it will almost surely converge on just that small set of theories with highest posterior probability (Geman and 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 up (see Fig. 1) which effectively results in lower-temperature search. Might children's thinking also evolve like this, for analogous functional reasons (Ullman et al., 2012; Ullman, 2015; Gopnik et al., 2015, 2017)? Whether through an explicit temperature parameter and annealing schedule that injects more randomness into younger children's thoughts and actions, or simply 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 (Schulz, 2012a; Chu and Schulz, 2020). To paraphrase Schulz, 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 rain-drops inside the clouds are jumping up and down'. But these proposals are at least relevant, systematically related to the phenomena to be explained, and at least potentially correct explanations. In contrast, an answer such as 'The clouds are bigger than ladybugs' or 'You can eat raw cheese' wouldn't be false, but it would not occur to a child in this context (except as a joke or miscommunication) because it's just 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, and thereby make the problem of theory search far more constrained and tractable (Schulz, 2012a).

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, though a number of possibilities have been put forward recently. For example, model-free learning (Sutton and Barto, 2018; Phillips et al., 2019) and amortized inference, or learned data-driven proposals (Shi et al., 2010; Gershman and Goodman, 2014), 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 only generate certain hypotheses 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, maybe learning is 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, and specifically probabilistic generative programs in the case of learning intuitive theories. This view could help to 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. (in press) develop this idea under the name of "the child as hacker." Although it could be called "the child as coder/programmer/software-engineer" 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 most associate 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 and Griffiths, 2020). And other more aesthetic or communicative goals should be in play too: making code more reusable, elegant, understandable, clever, or simply more fun. Children's learning could be motivated by all these goals, and that means we need learning algorithms which can optimize for all these objectives.

Now consider those algorithms. Hackers deploy a diverse toolkit 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 a hierarchical Bayesian model with fixed structure. But most ways to improve programs require writing new code: This includes writing new functions, but also extending or debugging 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 "child as hacker/coder/software engineer" view 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 by building the concepts needed to understand and produce natural language (Landau, 1985; Waxman and Markow, 1995; Gopnik and Meltzoff, 1997), and then using language to build their knowledge through cultural or social means (Wellman, 2014; Gelman, 2009; Carey, 2009). 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 one 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 (Smith, 1984; Gulwani et al., 2017). 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 towards 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 (Smith, 1985; Alur et al., 2017), backwards chaining of goal-subgoal constraints or constraints on types of functions and the inputs they require (Polozov and Gulwani, 2015; Osera and Zdancewic, 2015) and higher-order program templates to guide search and form abstractions (Solar-Lezama, 2009; Lin et al., 2014; Cropper and Muggleton, 2017). 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; Ullman et al., 2018; Ellis et al., 2016; Rule et al., 2018; Ellis et al., 2018, 2020), and learning probabilistic generative programs (Hewitt et al., 2020; Ellis 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 work 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 gives 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 to the child-as-scientist. Formal scientific theories and intuitive theories both aim to solve the same problem: Finding generative programs that best account for a body of data (c.f., Li et al., 2008). Formal and intuitive theories both operate on multiple levels of abstraction (Kuhn, 2012; Wellman and 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 to 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 scientists are big children, Gopnik and 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 rationally re-uses 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, 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, their 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 sub-routines, variables, and data-formatting. 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 as well as the understanding of concepts others have discovered. In searching for concepts, the PLM will make proposals 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', c.f., Nersessian 1992 and Dirac, 1963). Even if such pictures don't 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 to 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 ($I = \frac{V}{R}$). It can be seen 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 just 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, consider if we told 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 mini-program any shorter or more predictive, and an ideal PLM could reject it. And yet, a 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 re-interpret 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), or 'T-cells don't want to harm healthy cells' (T-cells don't want anything), or even 'the electron wants to expand the sub-lattice' (electrons *can't* want anything).

We see then that 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, 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 give 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 time scales, with different experiences, using different primitives, under different constraints. There is much to explore in future work 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 just 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.

ACKNOWLEDGMENTS

We thank Fei Xu, Alison Gopnik, Henry Wellman, Charles Kemp, Amy Perfors, Noah Goodman, Vikash Mansinghka, 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 (CBMM) through NSF STC grant STC award CCF-1231216, Siegel Family Endowment, and the DARPA Machine Common Sense program.

References

Alur, R., Radhakrishna, A., Udupa, A., 2017. Scaling enumerative program synthesis via divide and conquer, in: Legay, A., T., M. (Eds.), Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2017), Lecture Notes in Computer Science vol 10205. Springer.

Anderson, E.M., Hespos, S.J., Rips, L.J., 2018. Five-month-old infants have expectations for the accumulation of nonsolid substances. Cognition 175, 1–10.

Anderson, J.R., 1990. The adaptive character of thought. Erlbaum, Hillsdale, NJ.

Baillargeon, R. abd Li, J., Gertner, Y., Wu, D., 2011. How do infants reason about physical events?. 2nd ed.. Blackwell. pp. 11–48.

Baillargeon, R., Li, J., Ng, W., Yuan, S., 2009. An account of infants' physical reasoning. Oxford University Press.

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

Baker, C.L., Jara-Ettinger, J., Saxe, R., Tenenbaum, J.B., 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. Nature Human Behaviour 1, 0064.

Baker, C.L., Saxe, R., Tenenbaum, J.B., 2009. Action understanding as inverse planning. Cognition 113, 329 – 349.

Balog, M., Gaunt, A.L., Brockschmidt, M., Nowozin, S., Tarlow, D., 2017. Deepcoder: Learning to write programs. ArXiv abs/1611.01989.

Bates, C.J., Yildirim, I., Tenenbaum, J.B., Battaglia, P., 2019. Modeling human intuitions about liquid flow with particle-based simulation. PLoS computational biology 15, e1007210.

Battaglia, P.W., Hamrick, J.B., Tenenbaum, J.B., 2013. Simulation as an engine of physical scene understanding. Proceedings of the National Academy of Sciences 110, 18327–18332.

Baum, E.B., 2004. What is thought? MIT press.

Bonawitz, E., Ullman, T.D., Bridgers, S., Gopnik, A., Tenenbaum, J.B., 2019. Sticking to the evidence? a behavioral and computational case study of micro-theory change in the domain of magnetism. Cognitive science 43, e12765.

Carey, S., 1985. Conceptual change in childhood. MIT Press/Bradford Books, Cambridge, MA.

Carey, S., 2009. The Origin of Concepts. Oxford University Press.

Carey, S., Spelke, E., 1994. Domain-specific knowledge and conceptual change. Cambridge University Press. pp. 169–200.

Carey, S., Spelke, E., 1996. Science and core knowledge. Philosophy of Science 63, 515–533.

Chater, N., Oaksford, M., 2013. Programs as causal models: Speculations on mental programs and mental representation. Cognitive Science 37, 1171–1191.

Chu, J., Schulz, L.E., 2020. Play, curiosity, and cognition. Annual Review of Developmental Psychology .

Craik, K., 1943. The Nature of Explanation. Cambridge University Press.

Cropper, A., Muggleton, S.H., 2017. Learning higher-order logic programs through abstraction and invention. International Joint Conference on Artificial Intelligence (IJCAI) , 1418–1424.

Csibra, G., 2008. Goal attribution to inanimate agents by 6.5-month-old infants. Cognition 107, 705 – 717.

Csibra, G., Biró, S., Koós, O., Gergely, G., 2003. One-year-old infants use teleological representations of actions productively. Cognitive Science 27, 111–133.

Czégel, D., Zachar, I., Szathmáry, E., 2018. Major evolutionary transitions as bayesian structure learning. bioRxiv , 359596.

Dechter, E., Malmaud, J., Adams, R.P., Tenenbaum, J.B., 2013. Bootstrap learning via modular concept discovery, in: Twenty-Third International Joint Conference on Artificial Intelligence.

Devlin, J., Uesato, J., Bhupatiraju, S., Singh, R., rahman Mohamed, A., Kohli, P., 2017. Robustfill: Neural program learning under noisy i/o, in: ICML.

Dewar, K.M., Xu, F., 2010. Induction, overhypothesis, and the origin of abstract knowledge: Evidence from 9-month-old infants. Psychological Science 21, 1871–1877.

Dirac, P.A.M., 1963. The evolution of the physicist's picture of nature. Scientific American 208, 45–53.

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, pp. 7805–7815.

Ellis, K., Solar-Lezama, A., Tenenbaum, J., 2016. Sampling for bayesian program learning, in: Advances in Neural Information Processing Systems, pp. 1297–1305.

Ellis, K., Wong, C., Nye, M., Sablé-Meyer, M., Cary, L., Morales, L., Hewitt, L., Solar-Lezama, A., Tenenbaum, J.B., 2020. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. ArXiv abs/2006.08381.

Erdogan, G., Jacobs, R.A., 2017. Visual shape perception as bayesian inference of 3d object-centered shape representations. Psychological Review 124, 740—761.

Esposito, A., 1969. A simplified method for analyzing hydraulic circuits by analogy. Machine Design 41, 173.

Fodor, J.A., 1975. The language of thought. Harvard University Press: Cambridge, MA.

Fodor, J.A., 1998. Concepts: Where Cognitive Science Went Wrong. New York: Oxford University Press.

Forbus, K., 2019. Qualitative Representations: How People Reason and Learn about the Continuous World. The MIT Press, MIT Press.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2013. Bayesian data analysis. 3rd ed., Chapman & Hall, New York.

Gelman, S.A., 2009. Learning from others: Children's construction of concepts. Annual review of psychology 60, 115–140.

Geman, S., Geman, D., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence , 721–741.

Gentner, D., 1989. The mechanisms of analogical learning. Cambridge University Press. pp. 199–241.

Gentner, D., Stevens, A.L., 1983. Mental models. Psychology Press.

Gershman, S.J., Goodman, N.D., 2014. Amortized inference in probabilistic reasoning. Cognitive Science 36.

Gershman, S.J., Horvitz, E.J., Tenenbaum, J.B., 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. Science 349, 273–278.

Gerstenberg, T., Tenenbaum, J.B., 2017. Intuitive theories. Oxford handbook of causal reasoning , 515–548.

Gilden, D.L., Proffitt, D.R., 1989. Understanding collision dynamics. Journal of Experimental Psychology: Human Perception and Performance 15, 372.

Goodman, N., 1983. Fact, fiction, and forecast. Harvard University Press.

Goodman, N., Ullman, T., Tenenbaum, J., 2011a. Learning a theory of causality. Psychological Review 118, 110–119.

Goodman, N.D., Mansinghka, V.K., Roy, D.M., Bonawitz, K., Tenenbaum, J.B., 2008. Church: a language for generative models. Uncertainty in Artificial Intelligence .

Goodman, N.D., Tenenbaum, J.B., Contributors, T.P., 2016. Probabilistic Models of Cognition. `http://probmods.org/v2`. Accessed: 2020-6-28.

Goodman, N.D., Tenenbaum, J.B., Gerstenberg, T., 2014. Concepts in a probabilistic language of thought. Technical Report. Center for Brains, Minds and Machines (CBMM).

Goodman, N.D., Ullman, T.D., Tenenbaum, J.B., 2011b. Learning a theory of causality. Psychological Review 118, 110—119.

Gopnik, A., 1996. The scientist as child. Philosophy of Science 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. Psychological Review 111, 1–31.

Gopnik, A., Griffiths, T.L., Lucas, C.G., 2015. When younger learners can be better (or at least more open-minded) than older ones. Current Directions in Psychological Science 24, 87–92.

Gopnik, A., Meltzoff, A.N., 1997. Words, Thoughts, and Theories. MIT Press, Cambridge, MA.

Gopnik, A., O'Grady, S., Lucas, C.G., Griffiths, T.L., Wente, A., Bridgers, S., Aboody, R., Fung, H., Dahl, R.E., 2017. Changes in cognitive flexibility and hypothesis search across human life history from childhood to adolescence to adulthood. Proceedings of the National Academy of Sciences 114, 7892–7899.

Gopnik, A., Wellman, H., 1992. Why the child's theory of mind really is a theory. Mind and Language 7, 145–171.

Gopnik, A., Wellman, H., 1994. The theory theory. Mapping the mind: Domain specificity in cognition and culture , 257–293.

Gopnik, A., Wellman, H.M., 2012. Reconstructing constructivism: Causal models, bayesian learning mechanisms, and the theory theory. Psychological bulletin 138, 1085.

Gregory, J., 2018. Game engine architecture. crc Press.

Griffiths, T.L., Chater, N., Kemp, C., Perfors, A., Tenenbaum, J.B., 2010. Probabilistic models of cognition: exploring representations and inductive biases. Trends in Cognitive Sciences 14, 357 – 364.

Griffiths, T.L., Kemp, C., Tenenbaum, J.B., 2008. Bayesian Models of Cognition. Cambridge University Press. p. 59–100.

Griffiths, T.L., Lieder, F., Goodman, N.D., 2015. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. Topics in cognitive science 7, 217–229.

Griffiths, T.L., Tenenbaum, J.B., 2005. Structure and strength in causal induction. Cognitive Psychology 51, 285–386.

Griffiths, T.L., Tenenbaum, J.B., 2009. Theory-based causal induction. Psychological Review 116, 661–716.

Griffiths, T.L., Vul, E., Sanborn, A.N., 2012. Bridging levels of analysis for probabilistic models of cognition. Current Directions in Psychological Science 21, 263–268.

Gulwani, S., Hernández-Orallo, J., Kitzelmann, E., Muggleton, H.S., Schmid, U., Zorn, G.B., 2015. Inductive programming meets the real world. Communications of the ACM .

Gulwani, S., Polozov, A., Singh, R., 2017. Program Synthesis. volume 4. NOW.

Hamrick, J.B., Battaglia, P.W., Griffiths, T.L., Tenenbaum, J.B., 2016. Inferring mass in complex scenes by mental simulation. Cognition 157, 61–76.

Henderson, L., Goodman, N.D., Tenenbaum, J.B., Woodward, J.F., 2010. The structure and dynamics of scientific theories: A hierarchical bayesian perspective. Philosophy of Science 77, 172–200.

Hewitt, L.B., Le, T.A., Tenenbaum, J.B., 2020. Learning to learn generative models with memoized wake-sleep. Uncertainty in Artificial Intelligence (UAI) .

Jara-Ettinger, J., Gweon, H., Schulz, L.E., Tenenbaum, J.B., 2016. The naïve utility calculus: Computational principles underlying commonsense psychology. Trends in cognitive sciences 20, 589–604.

Jaynes, E.T., 2003. Probability theory: The logic of science. Cambridge University Press, Cambridge.

Johnson-Laird, P.N., 2004. The history of mental models, in: Psychology of reasoning. Psychology Press, pp. 189–222.

Johnston, A.M., Sheskin, M., Keil, F.C., 2019. Learning the relevance of relevance and the trouble with truth: Evaluating explanatory relevance across childhood. Journal of Cognition and Development 20, 555–572.

Kemp, C., 2008. The acquisition of inductive constraints. Ph.D. thesis. Massachusetts Institute of Technology.

Kemp, C., Goodman, N.D., Tenenbaum, J.B., 2008. Theory acquisition and the language of thought, in: Proceedings of Thirtieth Annual Meeting of the Cognitive Science Society.

Kemp, C., Goodman, N.D., Tenenbaum, J.B., 2010. Learning to learn causal models .

Kemp, C., Perfors, A., Tenenbaum, J., 2007. Learning overhypotheses with hierarchical Bayesian models. Developmental Science 10, 307–321.

Kemp, C., Tenenbaum, J.B., 2008. The discovery of structural form. Proceedings of the National Academy of Sciences 105, 10687–10692.

Kemp, C., Tenenbaum, J.B., 2009. Structured statistical models of inductive reasoning. Psychological Review 116, 20 – 58.

Kiley Hamlin, J., Ullman, T., Tenenbaum, J.B., Goodman, N., Baker, C., 2013. The mentalistic basis of core social cognition: experiments in preverbal infants and a computational model. Developmental science 16, 209–226.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. science 220, 671–680.

Koza, J.R., et al., 1994. Genetic programming II. volume 17. MIT press Cambridge.

Kubricht, J., Zhu, Y., Jiang, C., Terzopoulos, D., Zhu, S., Lu, H., 2017. Consistent probabilistic simulation underlying human judgment in substance dynamics, in: Gunzelmann, G., Howes, A., Tenbrink, T., Davelaar, E.J. (Eds.), Proceedings of the 39th Annual Meeting of the Cognitive Science Society.

Kuhn, T.S., 2012. The structure of scientific revolutions. University of Chicago press.

Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B., 2015. Human-level concept learning through probabilistic program induction. Science 350, 1332–1338.

Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J., 2017. Building machines that learn and think like people. Behavioral and Brain Sciences 40.

Landau, Barbara an d Gleitman, L., 1985. Language and Experience: Evidence from the Blind Child. Harvard University Press.

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

Levin, L.A., 1973. Universal sequential search problems. Problemy Peredachi Informatsii 9, 115–116.

Li, M., Vitányi, P., et al., 2008. An introduction to Kolmogorov complexity and its applications. volume 3. Springer.

Lieder, F., Griffiths, T.L., 2020. Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. Behavioral and Brain Sciences 43.

Lin, D., Dechter, E., Ellis, K., Tenenbaum, J.B., Muggleton, S., 2014. Bias reformulation for one-shot function induction. European Conference on Artificial Intelligence (ECAI) .

Liu, S., Ullman, T.D., Tenenbaum, J.B., Spelke, E.S., 2017. Ten-month-old infants infer the value of goals from the costs of actions. Science 358, 1038–1041.

Lombrozo, T., 2016. Explanatory preferences shape learning and inference. Trends in Cognitive Sciences 20, 748–759.

Lucas, C.G., Griffiths, T.L., Xu, F., Fawcett, C., Gopnik, A., Kushnir, T., Markson, L., Hu, J., 2014. The child as econometrician: A rational model of preference understanding in children. PloS one 9, e92160.

Mach, E., 1910. Populär-wissenschaftliche Vorlesungen. Barth.

Marr, D., 1982. Vision. Freeman Publishers.

Murphy, G.L., Medin, D.L., 1985. The role of theories in conceptual coherence. Psychological Review 92, 289–316.

Nersessian, N.J., 1992. How do scientists think? capturing the dynamics of conceptual change in science. Cognitive models of science 15, 3–44.

Nye, M., Hewitt, L., Tenenbaum, J., Solar-Lezama, A., 2019. Learning to infer program sketches, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, PMLR. pp. 4861–4870.

Osera, P.M., Zdancewic, S., 2015. Type-and-example-directed program synthesis, in: Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, New York, NY, USA. p. 619–630.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: Networks of plausible inference .

Pearl, J., 2000. Causality: models, reasoning, and inference. Cambridge University Press.

Perfors, A., Tenenbaum, J.B., Griffiths, T.L., Xu, F., 2011. A tutorial introduction to Bayesian models of cognitive development. Cognition 120, 302–321.

Perfors, A., Tenenbaum, J.B., Wonnacott, E., 2010. Variability, negative evidence, and the acquisition of verb argument constructions. Journal of Child Language 37, 607–642.

Phillips, J., Morris, A., Cushman, F., 2019. How we know what not to think. Trends in cognitive sciences .

Piantadosi, S.T., Tenenbaum, J.B., Goodman, N.D., 2012. Bootstrapping in a language of thought: A formal model of numerical concept learning. Cognition 123, 199–217.

Piantadosi, S.T., Tenenbaum, J.B., Goodman, N.D., 2016. The logical primitives of thought: Empirical foundations for compositional cognitive models. Psychological review 123, 392–424.

Polozov, O., Gulwani, S., 2015. Flashmeta: A framework for inductive program synthesis, in: OOPSLA 2015 Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, pp. 107–126.

Rule, J., Schulz, E., Piantadosi, S.T., Tenenbaum, J.B., 2018. Learning list concepts through program induction, in: Proceedings of the 40th Annual Conference of the Cognitive Science Society, Cognitive Science Society.

Rule, J.S., Tenenbaum, J.B., Piantadosi, S.T., in press. The child as hacker. Trends in Cognitive Sciences .

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

Saad, F.A., Cusumano-Towner, M.F., Schaechtle, U., Rinard, M.C., Mansinghka, V.K., 2019. Bayesian synthesis of probabilistic programs for automatic data modeling. Proc. ACM Program. Lang. 3, 37:1–37:32.

Sanborn, A.N., Mansinghka, V.K., Griffiths, T.L., 2013. Reconciling intuitive physics and newtonian mechanics for colliding objects. Psychological review 120, 411.

Schulz, L., 2012a. Finding new facts; thinking new thoughts, in: Advances in child development and behavior. Elsevier. volume 43, pp. 269–294.

Schulz, L., 2012b. The origins of inquiry: Inductive inference and exploration in early childhood. Trends in cognitive sciences 16, 382–389.

Shi, L., Griffiths, T.L., Feldman, N.H., Sanborn, A.N., 2010. Exemplar models as a mechanism for performing bayesian inference. Psychonomic Bulletin and Review 17, 443–464.

Siegler, R.S., 2007. Cognitive variability. Developmental science 10, 104–109.

Siegler, R.S., Chen, Z., 1998. Developmental differences in rule learning: A microgenetic analysis. Cognitive Psychology 36, 273–310.

Simon, H.A., 1956. Rational choice and the structure of the environment. Psychological review 63, 129.

Simon, H.A., 1962. An information processing theory of intellectual development. Monographs of the Society for Research in Child Development 27, 154–155.

Smith, D.R., 1984. Synthesis of lisp programs from examples: A survey, in: A.W. Biermann, A.W., Guiho, G., Kodratoff, Y. (Eds.), Automatic Program Construction Techniques. MacMillan Publishing Company, pp. 307–324.

Smith, D.R., 1985. Top-down synthesis of divide-and-conquer algorithms. Artificial Intelligence 27, 43–96.

Smith, K., Mei, L., Yao, S., Wu, J., Spelke, E., Tenenbaum, J.B., Ullman, T., 2019. Modeling expectation violation in intuitive physics with coarse probabilistic object representations, in: Advances in Neural Information Processing Systems, pp. 8983–8993.

Smith, K.A., Mei, L., Yao, S., Wu, J., Spelke, E., Tenenbaum, J.B., Ullman, T.D., 2020. The fine structure of surprise in intuitive physics: When, why, and how much?, in: Proceedings of the 42nd Annual Meeting of the Cognitive Science Society.

Solar-Lezama, A., 2009. The sketching approach to program synthesis, in: Hu, Z. (Ed.), Programming Languages and Systems, 7th Asian Symposium, APLAS 2009, Seoul, Korea, December 14-16, 2009. Proceedings, Springer. pp. 4–13.

Spelke, E., 1990. Principles of object perception. Cognitive Science 14, 29–56.

Spelke, E., 1995. Initial knowledge: Six suggestions. Cognition on cognition , 433–447.

Spelke, E., Kinzler, K., 2007. Core knowledge. Developmental Science 10, 89–96.

Spelke, E.S., Breinlinger, K., Macomber, J., Jacobson, K., 1992. Origins of knowledge. Psychological Review 99, 605 – 632.

Spelke, E.S., Kestenbaum, R., Simons, D.J., Wein, D., 1995. Spatiotemporal continuity, smoothness of motion and object identity in infancy. British Journal of Developmental Psychology 13, 113–142.

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

Stanley, K.O., Clune, J., Lehman, J., Miikkulainen, R., 2019. Designing neural networks through neuroevolution. Nature Machine Intelligence 1, 24–35.

Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

Tenenbaum, J.B., Griffiths, T.L., Kemp, C., 2006. Theory-based Bayesian models of inductive learning and reasoning. Trends in Cognitive Sciences 10, 309–318.

Tenenbaum, J.B., Griffiths, T.L., Niyogi, S., 2007. Intuitive theories as grammars for causal inference, in: Gopnik, A., Schulz, L. (Eds.), Causal learning: Psychology, philosophy, and computation. Oxford University Press, Oxford.

Tenenbaum, J.B., Kemp, C., Griffiths, T.L., Goodman, N.D., 2011. How to grow a mind: Statistics, structure, and abstraction. Science 331, 1279–1285.

Thagard, P., 2008. Conceptual change in the history of science: Life, mind, and disease. International handbook of research on conceptual change , 374–387.

Turing, A.M., 2009. Computing machinery and intelligence, in: Parsing the Turing Test. Springer, pp. 23–65.

Téglás, E., Vul, E., Girotto, V., Gonzalez, M., Tenenbaum, J.B., Bonatti, L.L., 2011. Pure reasoning in 12-month-old infants as probabilistic inference. Science 332, 1054–1059.

Ullman, T., Siegel, M.H., Tenenbaum, J.B., Gershman, S., 2016. Coalescing the vapors of human experience into a viable and meaningful comprehension., in: CogSci.

Ullman, T.D., 2015. On the nature and origin of intuitive theories: Learning, physics and psychology. Ph.D. thesis. Massachusetts Institute of Technology.

Ullman, T.D., Goodman, N.D., Tenenbaum, J.B., 2012. Theory learning as stochastic search in the language of thought. Cognitive Development 27, 455–480.

Ullman, T.D., Spelke, E., Battaglia, P., Tenenbaum, J.B., 2017. Mind games: Game engines as an architecture for intuitive physics. Trends in Cognitive Sciences 21, 649–665.

Ullman, T.D., Stuhlmüller, A., Goodman, N.D., Tenenbaum, J.B., 2018. Learning physical parameters from dynamic scenes. Cognitive Psychology 104, 57–82.

Vul, E., Goodman, N., Griffiths, T.L., Tenenbaum, J.B., 2014. One and done? optimal decisions from very few samples. Cognitive science 38, 599–637.

Waxman, S.R., Markow, D.B., 1995. Words as invitations to form categories: Evidence from 12- to 13-month-old infants. Cognitive Psychology 29, 257–302.

Wellman, H., Gelman, S., 1998. Knowledge acquisition in foundational domains. Wiley. volume 2. pp. 523–573.

Wellman, H.M., 2014. Making minds: How theory of mind develops. Oxford University Press.

Wellman, H.M., Fang, F., Peterson, C.C., 2011. Sequential progressions in a theory-of-mind scale: Longitudinal perspectives. Child development 82, 780–792.

Woodward, A., Needham, A., 2008. Learning and the infant mind. Oxford University Press.

Woodward, A.L., 1998. Infants selectively encode the goal object of an actor's reach. Cognition 69, 1–34.

Woodward, A.L., 1999. Infants' ability to distinguish between purposeful and non-purposeful behaviors. Infant Behavior and Development 22, 145–160.

Wu, J., Yildirim, I., Lim, J.J., 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, pp. 127–135.

Xu, F., 2019. Towards a rational constructivist theory of cognitive development. Psychological review 126, 841.

Xu, F., Tenenbaum, J.B., 2007. Word learning as bayesian inference. Psychological Review .