Learning a Theory of Mind

Isaac Davis

Contents

1 Introduction and overview

1.1 Theory of Mind and Cognitive Development

Human beings are uniquely social animals. We are born into and raised in social communities, and as we develop, we come to understand and explain those around us in terms of hidden psychological states. The term Theory of Mind (ToM) encompasses our ability to make these psychological inferences, and while we are by no means the only social species, the capacity to reason about others' psychological states is apparently unique to humans (Penn et al 2008). This ability is crucial for language use, social interaction, and social cognition, and the notion of hidden psychological states is deeply entrenched in our language and cognition. Children begin to describe people in terms of psychological states almost as soon as they learn to speak at all (Gopnik & Slaughter 1991). Even in explaining the behavior of non-human objects, children often appeal to

mentalistic metaphors to fill in conceptual gaps (e.g. the elementary-school physics explanation that an electron *wants* to be in a low-energy state).

In recent years, there has been a surge of interest in understanding how our ToM develops; that is, how do we acquire, store, and manipulate the knowledge encompassed by ToM? Part of this interest is driven by the notion our ToM might constitute a foundational human cognition- a core cognitive system that serves as a fundamental building block for more complex cognitive capacities (Wellman 2014). Another reason for this interest is that the nature of our developmental data on ToM is somewhat challenging to explain. On the one hand, we have a significant (and steadily growing) body of psychological data demonstrating clear and consistent developmental steps in human ToM acquisition, which seem to universally culminate in a "belief-desire-action" (BDA) heuristic: roughly, we expect people to *do* what they *believe* will get them what they *want*. While there is a moderate degree of cross-cultural variability in the ages at which children acquire certain ToM-skills, there is a very a low degree of cross-cultural variability in the order in which children tend to acquire those skills. On the other hand, while the developmental patterns tend to be cross-culturally very consistent, there are notable exceptions which are tightly correlated to specific cultures, native languages, or other social contingencies. This makes developmental data on ToM difficult to explain from a strictly nativist or associationist perspective.

1.2 Contributions of this thesis

The goal of this thesis is to provide the foundations for a rationalist account of human ToM and its development. We focus on the nature, structure, and origin of the

"belief-desire-action" (BDA) heuristic that underlies human psychological explanations. In particular, we shall argue that

1. certain features of BDA folk psychologies are characteristic of rational solutions to the social inference problems we face in our everyday social experience,

2. it is *plausible* that we develop BDA folk psychologies *because* they are rational solutions to the social inference problems we face in our everyday social experience,

3. to the degree that (2) is true, the developmental trajectories (and deviations) we observe in our own ToM development may reflect inductive stages of rational inference over dynamic social data.[1]

Our approach in developing these arguments is strongly motivated by a "rational constructivist" or "Theory-theory" perspective. According to this view, human beings learn about the world through a process much like scientific inference, and store this knowledge in the form of "intuitive theories," which share important structural and dynamic features with scientific theories (Piaget 1964, Gopnik et al 1997). To formalize this theoretical model of "cognitive development as rational inference," we construct a computational modeling framework with three core components:

1. A formal characterization of the behavioral and psychological inference problems we face in our everyday social experience. This includes the data we collect in social environments (observation of and interaction with other agents), the tasks

[1]The stronger argument that our developmental trajectories *do*, in fact, reflect rational inference requires empirical analysis that is beyond the scope of this dissertation. We do, however, outline a methodology for performing this analysis in chapter 4.

that we perform while navigating social environments (predicting, explaining, and reasoning about other agents), and the cognitive constraints under which we operate.

2. A formal characterization of possible solutions to those problems (i.e. possible theories of mind and psychological explanations), and a normative principle of rationality for evaluating those solutions.

3. An inference mechanism (ideally a domain-general one) which leverages this principle of rationality to *learn* how to solve social inference problems.

This will allow us to precisely characterize and distinguish "BDA-like" folk psychologies, demonstrate (analytically and through simulation) in what sense they are rational solutions to social inference problems, and demonstrate how (and under what conditions) they could be learned from data through rational inference. In this sense, we shall argue that the Belief-Desire-Action heuristic which underlies human psychological explanation may emerge naturally due to more general principles of pragmatically rational inference.

1.3 Outline

The remainder of this thesis is structured as follows. In chapter 2, we provide the scientific, philosophical, and conceptual background for our project. We start by reviewing the relevant developmental data in more detail, and explaining some of the main theoretical challenges in accounting for this data. We then describe the main conceptual approaches that have been used in explaining this cognitive development, and outline our "constructivist" perspective in more detail. This motivating perspective- that

our folk psychology is an intuitive theory acquired through an approximately rational inference process- will dictate our general approach to these questions: First we identify the inference problems we face in the social domain, a space of plausible solutions to those problems (given certain assumptions about our cognitive constraints), and a constraint-sensitive rationality principle for evaluating potential solutions in light of social data. We then formally characterize the defining features of BDA folk psychologies, and in what sense (and under what circumstance) those features are rational.

In chapter 3, we present a computational modeling framework for inferring, reasoning with, and evaluating psychological explanations of agent behavior. The nature of our query requires that we represent learning and inference at multiple levels of abstraction. We draw on the formal machinery of Bayesian Theory of Mind, Hierarchical Bayesian Models, and Probabilistic Programs to develop this framework. Importantly, we define our framework at an algorithmic, rather than computational, level of analysis, as we will need algorithmic-level information to derive a rationality principle that is sensitive to the learner's cognitive constraints.

In chapter 4, we present a methodological framework for connecting these computational models to behavioral data in a scientific context. We focus in particular on infant cognitive-behavioral studies, which are particularly challenging due in part to the extreme sparsity of available data. We briefly review the most common paradigms for infant cognitive studies, identify a set of related conceptual and methodological challenges inherent to these studies, and illustrate how our modeling framework can be used to help resolve these challenges. Finally, we present a case study in which we use our modeling framework to provide a more refined interpretation of behavioral data from a previous study on infant goal attribution (Woodward 1998).

In chapter 5, we demonstrate how our framework provides a theoretical basis for studying and modeling ToM development via simulation. We first define and justify a set of assumptions about the learner's cognitive constraints, and draw on principles from philosophy of science and epistemology to derive a normative rationality principle in terms of these constraints. We then demonstrate how certain features characteristic of our commonsense psychological explanations can be defined as representational and relational constraints on models. This allows us to identify and distinguish "BDA-like" folk psychologies in the context of our framework. We then demonstrate, using both analytic arguments and simulations, how and under what circumstances certain features of BDA folk psychologies constitute rational solutions to social inference problems. Finally, we draw on the highest (i.e. most abstract) level of our hierarchical model to demonstrate how this rationality principle can be leveraged to *learn* a theory of mind from social data, and how this may help us explain our own cognitive development. In the final chapter, we outline a framework for evaluating these theoretical claims with empirical data, and describe several promising future directions for research.

2 Background and motivation

2.1 Developmental trajectories in ToM

It is difficult to exhaustively characterize every skill that ToM comprises, so we shall focus on certain developmental patterns that appear consistently in psychological data on ToM acquisition. We start by reviewing general developmental trends, before looking more specifically at developmental transitions that appear in children's understanding of

a) desires, goals, and intentions, and b) awareness, perceptions, and beliefs.

2.1.1 General timeline

It is well documented that very young infants, even newborns, preferentially attend to human faces, voices, and biomechanical movements over other stimuli, though there is some decline in this tendency throughout the first year (Farroni et al 2002, Frank et al 2009, Johnson et al 1991). While this does not show that infants have a well-formed concept of agency or attribute agency to other people, it does at least show that infants can, to some degree, discriminate between human-like and non-human-like objects. By 10-14 months, infants can consistently track an adult's eye-gaze to an object of interest, detect when an adult has or lacks visual access to an object, and understand, at least to some degree, the relation between visual access and awareness (Brooks & Meltzoff 2005, Kovács et al 2010, Onishi & Baillargeon 2005).

Around this same period, infants begin to identify and discriminate intentional or goal-directed actions. There is evidence that infants as young as 9 months can distinguish intentional from unintentional actions and adjust their reaction to an event based on this distinction (Behne et al 2005). Additionally, 9-12 month old infants can identify an actor's goal from repeated behavior and form expectations about the actor's future behavior accordingly (Gergely et al 1995, Phillips & Wellman 2005, Woodward 1998). Older infants (18-20 months) have been shown to infer preferences from statistical information, and leverage these inferred preferences when responding in controlled experiments (Kushnir et al 2010). Around 2 years old, children can comprehensively explain and predict behavior in terms of desires and goal satisfaction (Wellman & Woolley 1990). Additionally, children develop a partial understanding of how visual

awareness constrains behavior somewhere between 2 and 3 years old (Dunham et al 2000, O'Neil 1996). However, children 3 and under consistently struggle in tasks that require a representational understanding of beliefs, or the ability to explain erroneous behavior in terms of false beliefs (Wimmer & Perner 1983). This ability emerges consistently between 4 and 6 years of age, at which point children can comprehensively and (mostly) correctly explain and predict human behavior in terms of beliefs and desires (Perner & Wimmer 1985). Importantly, while the ages at which these skills emerge varies cross-culturally, the order in which skills are acquired appears to be very cross-culturally consistent (with a few notable exceptions) (Wellman & Liu 2004, Shahaeian et al 2011).

The *belief-desire-action* (BDA) framework that children develop around the age of 5 enables a wide range of mental and behavioral inferences, which closely mirror adult performance in many social inference tasks. These three categories- beliefs, desires, and actions- constitute the basis of human psychological explanations, which follow the same basic heuristic: in short, we expect people to *do* what they *believe* will get what they *want*. The full picture is, of course, more complicated, but this general heuristic appears to be nearly universal, emerging in typically developing children around 5-6 years of age. ToM does continue to develop past these ages; children develop a richer and more complex understanding of emotions (e.g. that one can experience contradictory emotions at the same time), an understanding of more abstract principles (e.g. the mind-brain distinction), and an appreciation for the constant flow of conscious mental activity that other agents engage in (Gnepp 1983, Flavell et al 1995). For our purposes, however, we are primarily interested in the development of the belief-desire-action explanatory framework which consistently emerges around 5-6 years of age.

2.1.2 Goals, desires, and intentional actions

As discussed in the previous section, infants begin to display some understanding of goals and intentional behavior around 10-12 months old, and 2 year old children can consistently explain and predict behaviors in terms of desires. However, it is clear that these abilities do not emerge all at once. The data reveal several stark transitions in children's understanding of goals and intentions, which we review in greater detail here.

Action-understanding in infants: outcomes, goals, and rationality An early demonstration of goal-awareness in infants comes from a 1998 study by Amanda Woodward (Woodward 1998). In this study, infants were repeatedly shown an event in which an actor reached for and grasped one of two visually distinct toys. The toys were placed such that one was closer and one further from the actor, so that reaching for one toy resulted in a distinct physical arm motion than reaching for the other (see figure 2.1.1). After being habituated to this repeated stimulus, the positions of the two toys were switched, and the infant was shown two test events. In the *new goal* test event, the actor performed the same physical reaching motion as in the habituation event, thereby grasping the opposite toy. In the *new motion* event, the actor reached for the same toy as in habituation, which required a different physical reaching motion. Infants as young as 8 months old were consistently more "surprised" by the *new goal* event than the *new motion* event. Under a standard interpretation, this suggests that infants encoded the actor's reach in terms of the target object rather than the spatiotemporal profile of the reach itself[2]. Thus, infants as young as 8-months old appear to be sensitive to the target

[2]See chapter 4 for a more detailed explanation of infant cognitive studies and the visual habituation paradigm

object of an actor's reach, and subsequently form expectations about the actor's future behavior in a new environment in terms of this target.

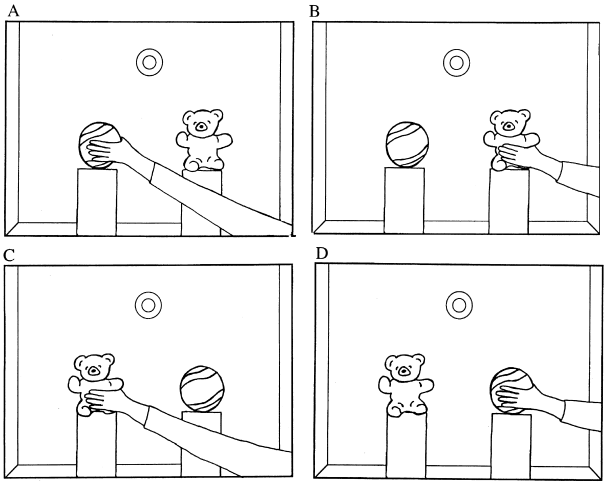


Figure 2.1.1: The four conditions used for habituation and test trials in Woodward (1998)

While this reveals that infants have some understanding of how *outcomes* relate to goals and actions, our adult understanding of goal-directed behavior allows us to predict not only *what* outcome an actor will seek but also *how* an actor will try to realize that outcome. In general, given two possible paths to a goal, we expect an agent to take the shorter/easier/less costly of those two paths, unless there is some compelling reason to take the longer path (e.g. the shorter path is obstructed). This general idea corresponds to what is sometimes called a "principle of rationality," and a 1995 study (Gergely et al 1995) demonstrated that infants form expectations consistent with a rational interpretation of goal-directed behavior.

In these studies, infants were habituated to one of two events in which an animated agent moved in an arced path towards a target object (see figure 2.1.2). In the "rational" condition, the direct path from start to target was blocked by a barrier, thus

necessitating the arced path. In the "nonrational" condition, the scene contained no barrier, and the agent traveled in the longer arced path *despite* the availability of the shorter, direct path. After habituation, the barrier was removed, and the infants were shown two test events: one in which the agent takes a direct path to the target object, and one in which the agent continues to travel in an arced path. The resulting data showed that infants habituated to the "rational" condition were consistently more surprised when the agent traveled in an arced path, while infants in the "nonrational" condition displayed no strong preference between either test event. These results indicate that infants interpret a behavior as goal-directed not only when the outcome is identical across multiple habituation events, but only when the identical outcome is achieved through "rational" means. When the agent takes an indirect path to the target even though a direct path is available, infants are less likely to interpret the agent's behavior as goal-directed, and are less surprised when the agent again chooses the indirect path in the test event. This result was been subsequently replicated and confirmed in Phillips & Wellman (2005), who additionally demonstrated that when the target object is removed, infants do not interpret the behavior as goal directed in either condition. That is, Philips & Wellman demonstrate that a salient target object is necessary for infants to apply a goal-based interpretation of behavior, and it is not sufficient to simply observed a repeated behavior.

The above data (and related extensions/replications) indicate that, by 10-12 months old, infants a) infer an agent's goal or target outcome from the *equifinality* of that outcome across multiple events, and b) expect a goal-directed agent to take the shortest available path to the target outcome, consistent with a "principle of rational action." However, it is not clear what internal representations and operations drive this behavior.

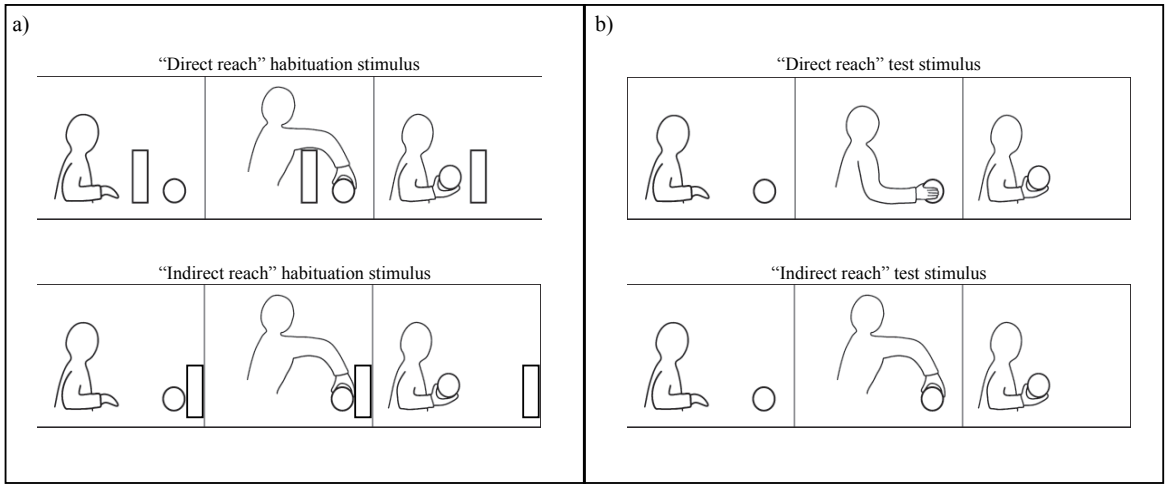


Figure 2.1.2: The two habituation conditions (panel a) and two test stimuli (panel b) used in Gergely et al 1995

One possibility is that infants possess a representational understanding of goal-directed behavior consistent with adult faculties. Under this *rich* account, infants attribute richly structured "desire" states to other agents, and predict behavior by inverting a "rational plan" for achieving the desired states given contemporaneous environmental constraints (Baker et al 2009). An alternative "lean" account is that infants do not possess anything resembling an adult understanding of goal-directed behavior, and are able to form these expectations solely from frequency information about observable features (Paulus et al 2011). A third possibility, sometimes called the "teleological" account, is that infants interpret intentional actions by relating outcomes, targets, and constraints through a principle of rational action, but do so without attributing explicit psychological states to the agent (Csibra & Gergely 1998). Under this account, the infant's "teleological" understanding of actions provides a conceptual scaffolding which supports the subsequent development of a representational understanding of action.

There are several experimental results which support the notion that a fundamental transition occurs in infants' understanding of intentional actions, and that this transition somehow involves the attribution of non-physical goal states which may differ from observed outcomes. Brandone & Wellman (2009) replicate the Gergely et al (1995) studies, but add an extra *unfulfilled goal* condition in which the actor *fails* to realize the target outcome (i.e. reaches over the barrier and attempts to grasp the object, but drops it as they pull their hand back). The authors found that 14- and 18- month old consistently form the same expectations in the *unfulfilled goal* condition as in the *fulfilled goal* condition (which is identical to the "rational" condition in the original experiment). That is, even when the actor's target outcome is not physically realized during the event, 14- and 18-month old infants are able to extrapolate the actor's intended outcome and predict behavior accordingly. However, 10-month old infants were unable to make that extrapolation in the *unfulfilled goal* case, and displayed the same reaction as infants in the original "nonrational" condition. That is, when the target outcome was *not* physically realized, 10-month old infants did not seem to interpret the actor's behavior as goal-directed. A similar difference was shown between 12- and 18month old infants in an active helping paradigm (Bellagamba & Tomasello 1999). In this study, infants were asked to help an adult participant complete a goal-directed action. In one condition, the infants observed the actor successfully realize the goal before being asked to do it themselves. In the other condition, the infants observed the actor attempt but fail to realize the same goal. The 18-month old infants consistently and accurately helped the actor achieve their goal regardless of whether or not they first observed a successful attempt, while 12-month old infants were mostly unable to replicate the goal in the case of an unsuccessful attempt. These results have been interpreted as evidence of an

intermediate stage in the development of children's action-understanding. Whereas younger infants need to see the goal physically realized in order to interpret behavior as goal-directed, older infants can extrapolate from failed attempts to correctly infer the actor's goal. This may suggest that younger infants dentify goals with the physical outcomes that *follow* actions, whereas older infants identify goals with a non-physical mental state that *precedes* (and/or *causes*) the action.

Action-understanding in older children: drives, desires, and intentions While data from infant studies do not clearly reveal when infants begin to attribute psychological states, it is clear from available data that, by two years of age, children possess some form of mentalistic understanding of human behavior. Two and three year olds can consistently explain and predict behavior in terms of what an agent "wants," and associate emotional reactions with the outcomes of goal-directed behavior. For example, when told that a puppet wants to have a snack and likes crackers the best, children as young as 2 will correctly predict the puppet's behavior (e.g. looking for crackers, choosing crackers over cereal) and emotional reaction (e.g. the puppet who finds crackers will be "happy," the puppet who fails to find crackers will be "sad") (Wellman & Wooley 1990).

While it is clear that two year olds regularly attribute and explain behavior in terms of desires, it is more difficult to discern how children understand the representational structure and functional/causal role of the mental states they attribute, what properties children associate with these states, and how this understanding develops throughout childhood. Two properties which appear fundamental to children's understanding of desires is that they are

- non-physical and internal to an agent,[3] and
- serve as some kind of motivating force for an agent's behavior. This is sometimes referred to as the "mind-to-world" property of desires (Searle & Willis 1983).

However, there are many different conceptualizations which could satisfy these basic properties, and the data reveal several apparent transitions in how children understand desires between 2 and 6 years old. To facilitate a discussion of that data, it will be helpful to distinguish between three related concepts. The first are *drives*, which comprise the most basic kind of internal motivation that might cause an agent to act. Having a drive entails some kind of internal attitude and may motivate an organism to act in a certain way (e.g. having an "itch" that compels the agent to scratch), but a drive, in and of itself, does not refer to anything outside of the agent, and depends only on the agent's own internal and/or bodily states. This contrasts with what are sometimes called "simple desires", which we may characterize as a drive "towards" an object or state of the world. For example, "thirst" is a drive, which describes a particular internal need or urge, and may be satiated in certain states of the world (e.g. a state in which the agent has had water). However, a "thirst state" on its own does not refer to a particular external state or object, even though it characterizes a need which is satiated in certain kinds of external states. By contrast, "wanting a glass of water" is a *desire*, as it consists of both a motivating attitude *and* an explicit reference to a target object or state. Thus, drives and desires both serve as internal causes of behavior, but only desires are "intentional" in the sense of referring to something outside the agent itself.

[3]Though Repacholi & Gopnik (1997) show that young infants may not fully recognize desires as "specific" to an agent, or that other agent's desires may differ from their own

Importantly, "simple desires" as conceived above are minimally representational. They consist of a target object or state and an attitude towards that state, but do not require additional representational structure beyond the physical target of a desire and the internal attitude associated with that target. Available data suggests that two and three year old children can competently predict an agent's behavior (or an agent's reaction to an outcome) by applying a straightforward "outcome matching" strategy: given an agent's stated desire, the 2 year old determines which action results in an outcome that most closely matches the stated desire, and predicts the agent will take that action; similarly, given the stated desire and subsequent outcome, the 2 year old determines whether the outcome matches the stated desire and predicts the agent's positive or negative emotional reaction accordingly (Wellman & Wooley 1990). However, two and three year old children seem to struggle with understanding the representational aspects of desires which are not directly tied to the target object. In one study, for example, 3 year olds struggled to understand that an agent's desires toward an object may change even if the object remains unchanged, while 4 year olds showed no such difficulty in the same tests (Gopnik & Slaughter 1991).[4]It has therefore been suggested that the 2-3 year old understanding of desires is largely non-representational, and that

[4]For example: a child is presented with 2 books, and asked which one they want to read. The experimenter then reads that book to the child, and then asks the child which book they want to read now. The child typically reports that now they want to read the second book. The child is then asked "which book did you want to read before? Which book do you want to read now?" 4 year olds consistently report both their previous desire and their new (changed) desire, while 3 year olds struggled to report their previous desire, and often responded as if they had wanted the second book all along.

the inability to understand representational mental states is also the reason for 3 year old childrens' difficulty in understanding beliefs (Gopnik & Wellman 1992).

The third important conceptual distinction is that between *desires* and *intentions*. While a desire may be a simple, non-representational state (consisting of a physical target and an attitude towards that target), an intention is more complex and explicitly representational. Intuitively, we may think of a desire as an attitude towards a certain object or outcome, and an intention as a structured plan to realize that desire (or more generally, to resolve a situation involving multiple, possibly conflicting desires, as well as beliefs and physical constraints). For example, my *desire to have a cookie* is distinct from my *intention to go get a cookie from the box in my kitchen* (and then eat it). The former consists only of the target state (one in which I have eaten a cookie) and a corresponding internal attitude (I would like to be in that state). The latter integrates my desire along with my relevant beliefs (e.g. that there is a box of cookies in my kitchen, that there are still cookies in the box, etc.) and any relevant physical constraints (I am in my living room and can easily walk to the kitchen) into a coherent plan to realize my desire.

Several studies reveal some stark differences in how 4 and 6 year old children conceptualize and distinguish between desires and intentions. In one study (Schult 2002), children were presented with carefully constructed stories in which desires and intentions diverge (e.g. I desire a cookie, I plan to go get one from my kitchen, but as I'm about to get up from my chair my roommate brings me a cookie; in this case my desire was fulfilled but my intention was not). In these studies, 3 and 4 year olds were consistently unable to distinguish desires from intentions, while 5 and 6 year olds frequently matched adult competencies in the same tests. Another study (Kushnir et al 2015) revealed a key difference in how 4 and 6 year olds understand the functional role of

desires and intentions. In particular, 4 year olds consistently described and reasoned about desires as direct causes of human behavior. For example, if I want a cookie and there is a cookie in front of me (which I am allowed to eat), then according to the 4 year old understanding, my desire to have a cookie *makes* me take the cookie. If I do not take the cookie, then, according to the 4 year old, there must be something preventing me from doing so. This contrasts with a 6 year old (and adult) understanding, according to which human beings may freely act against or inhibit their own desires. Thus, while the 4 year old seems to understand desires as a direct cause of human behavior, 6 year olds recognize desires as just one of several inputs that determine our specific intentions.

To briefly summarize these findings in a way most relevant to our current investigation:

1. Infants can form expectations about an agent's future behavior based on the equifinal structure of the agent's past behavior, but it is unclear whether (and at what stage) infants begin to attribute explicit psychological states to agents.

2. 2 and 3 year olds regularly attribute "simple desires" to agents, and can competently explain and predict behavior by matching stated desires with observed outcomes (or conversely, infer desires/preferences from repeated equifinal outcomes).

3. However, 2 and 3 year olds struggle to understand the representational aspects of desires not explicitly tied to the target state, for example that an agent's desires may change while the target object remains unchanged.

4. 4 year olds are more competent at understanding desires and changes in desires

that are not tied to a target object in an immediately apparent way. However, 4 year olds struggle to distinguish desires and intentions, and seem to interpret desires as direct causes of behavior, rather than one of several inputs that help form intentions. By 6 years old however, children seem to develop a representational understanding of intentions that largely approximates an adult understanding.

2.1.3 Perspective, awareness, and beliefs

Human infants appear to be born with an interest in human faces, voices, and eyes. Newborns will preferentially attend to human faces over other stimuli, and have been shown to distinguish direct eye contact from an averted gaze (Farroni et al 2002, Johnson et al 1991). The ability and tendency to follow an adult's line of sight to an object of interest- sometimes called "gaze-following"- consistently emerges by 6 months of age (D'Entremont et al 1997). A crucial question about early gaze-following is whether it reflects an understanding of the connection between visual access and knowledge/awareness. That is, do infants understand that "seeing is knowing," or does the infant's tendency to gaze-follow simply reflect the fact that following an adult's line of sight tends to reveal interesting information (i.e. whatever the adult is looking at)?

A number of experiments provide results which help discern these two possibilities. Infants 14-18 months old are significantly less likely to gaze-follow if the subject's eyes are closed, if the subject is wearing a blindfold, or if the subject's line of sight is visibly blocked by an opaque barrier. Infants 10-12 months old have more mixed results with the blindfold and barrier tests, but consistently distinguish between eyes-open and eyes-closed in gaze-following tests, and infants younger than 9 months appear not to discriminate between open and closed eyes when gaze-following (Brooks & Meltzoff 2002,

Brooks & Meltzoff 2005, Butler et al 2010). A 2008 study by Meltzoff & Brooks provides stronger evidence that 18 month old infants understand the connection between visual access and awareness. In this study, 18 month old infants were first given self-experience with a trick blindfold that appeared to be opaque from afar, but which did not occlude the subject's vision. After trying these blindfolds out for themselves, the infants were put through a similar test as in the above blindfold experiments. Infants who had experience with the trick blindfold were significantly more likely to gaze-follow a blindfolded adult than the control group who did not get to experience the trick blindfold. This suggests that, by 18 months, infant gaze-following is not purely driven by observable cues, and reflects at least a partial understanding of how visual access influences visual experience.

It is not always clear what specific features or cues drive infants' tendency to gaze-follow at different stages of development. It is clear, however, that older infants and young toddlers have some ability to a) determine an agent's visual access, b) determine, at least partially, how visual access influences awareness, and c) form expectations about an agent's behavior based on their awareness. Tomasello and Haberl (2003), for example, demonstrate that 12- and 18-month olds can determine what is "new" for other people. That is, if one of two adult participants is looking away while a new toy is introduced, 12- and 18- month old infants can recognize the "newness" of the toy to the adult that lacked visual access, and form corresponding expectations about that adult's behavior. Similarly, O'Neill (1996) demonstrates that, when requesting an adult's help in accessing an out-of-reach toy, 2 year olds will provide significantly more information and instruction when the adult did not see the toy being put away. Thus, it is clear that by 18-24 months, children have some understanding of how visual access influences knowledge and constrains behavior.

Note, however, that a child's performance in these tasks does not necessarily entail that the child attributes richly structured belief states or representations to agents. It is possible, for example, that younger children can only recognize whether or not an agent has epistemic access to an object, and how the lack of epistemic access to an object constrains that agent's behavior (relative to what the agent *would* have done if they *had* access). This sort of intermediate, "lean" access state would enable younger children to form expectations consistent with a rich understanding of belief in certain cases, but not others,[5] yet does not involve full fledged "belief" attributions.

Indeed, there is evidence that two year old children tend to struggle with tasks that require an understanding of the representational aspects of awareness. To this end, Masangkay et al (1974) distinguish between two kinds of perspective- and awareness-inference tasks. In a Level 1 task, a child must determine whether or not another agent has visual access to an object; for example, a 2 year old might be shown a card that contains a picture of a house on one side and is blank on the other. The card is then held so that the blank side faces the child while the other side faces an adult participant, and the child is then asked whether the adult can see the house. 2- and 3-year old children consistently pass such tests. Level 2 tasks, however, require an understanding of *how* an object appears to another agent. For example, the child might be shown a card with a picture of a house on one side and an arrow on the other, oriented so that if the arrow points down, the house is upside down. The card is then held so that the arrow faces the child and the house faces the adult, and the child is asked whether the house *appears* right-side up or upside-down to the adult. Children 4 and under consistently fail these tasks; the ability to consistently perform level 2 perspective inference emerges

[5]We provide a simulated example of this in chapter 5.3

around the ages of 5-6.

A second stark transition is revealed by the so-called "false-belief" test. In the archetypal false belief task (Wimmer & Perner 1983), the child is shown a video in which an actor ("Sally") places a toy into one of two boxes, then leaves the room. A second actor ("Anne") then enters the room and moves the toy to the other box. Finally, Sally re-enters the room and searches for her toy. The video is then paused and child is then asked to predict where Sally will search for her toy. For someone with an adult understanding of intentional behavior, the obvious answer is that Sally will search the box in which she initially hid the toy, as she was out of the room and therefore unaware of the toy being moved to the other box. In these tests, however, children 3 and under consistently answer that Sally will look in the box that actually contains the toy. The ability to correctly answer in these tests appears to emerge between the ages of 4 and 5.

These results have been cited as evidence that younger children lack an understanding of the representational aspects of belief- namely, that beliefs about the world may misrepresent or mismatch the actual state of the world. However, a number of studies have claimed to show an understanding of false beliefs in much younger children, even infants (Baillargeon et al 2010, Kovács et al 2010, Onishi & Baillargeon 2005). In these studies, variations of the standard false-belief test described above are translated into a non-verbal format compatible with the visual habituation paradigm. Infants as young as 10 months display dishabituation patterns consistent with a 5-6 year old's false belief understanding (e.g. fixating longer when the actor correctly searches the new box despite not observing the toy being moved to that box). These results present a conundrum: how do infants correctly understand how false beliefs constrain behavior when 3 year olds fail the verbal equivalents of the same tests? One proposal is that infants and young

toddlers track something belief-like but more primitive than a full fledged belief state. That is, infants can determine whether or not another agent is aware of an object (or aware of its location), and can infer how this awareness constrains action, but they do not attribute structured or representational "belief states" as older children do.

More generally, it has been suggested that difference between the 3-4 year old's ToM and the 5-6 year old's ToM is best characterized in terms of the "richness" or "leanness" of the attributed representations (Wellman & Wooley 1990). That is, the 3-4 year old ToM captures the motivating force and intentional nature of desires, the relation between visual access and awareness of one's surroundings, and the way that one's awareness constrains one's intentional behavior. The 5-6 year old's ToM extends this initial theory with richly structured representational mental states. This is further supported by the evidence of a transition between Level 1 and Level 2 perspective taking, where Level 1 tasks only require an understanding of "what" objects the agent is aware of, while Level 2 tasks require an understanding of "how" the objects appear to the agent.

2.1.4 Summary of key patterns and statement of goals

Before moving on, we will briefly recap these findings and relate them to our overall goals:

To recap: very young children and even infants (∼8-24 months) appear to understand the goal-directedness of intentional behavior, infer simple goals and preferences from observed behavior, and form expectations about future behavior based on inferred goals and a principle of rational action. Intuitively, children 2 and under expect an agent to take the *best possible* action which *fulfills their goals*. Additionally, children in this age range can determine whether another agent has visual access to an object or event,

understand how visual access constrains awareness, and understand how awareness constrains action. Intuitively, they can tell who sees what, understand the notion that "seeing is believing" (in some constrained sense), and understand how "lack of awareness of x" can inhibit behaviors normally generated by "desire for x." Children 3 and under consistently fail verbal false belief tests and tasks which require "Level 2" perspective taking. Both of these abilities tend to emerge around 4-5 years old. Younger children also struggle to distinguish intentions from desires, and seem to interpret desires as direct causes of action, whereas older children (5-6) distinguish the desire itself (i.e. a target object or state and a motivating attitude) from the intention to realize that desire through a specific means (i.e. the "plan" and planning process).

The data reveal a stark transition between the 2-year-old's ToM and the 5-year-old's. It has been argued that the fundamental distinction between the 2-year-old's ToM and the 5-year-old's is representation (Gopnik & Wellman 1992): the two year old, while capable of many complex social inferences, struggles with tasks that involve the representational aspects of other people's mental states. The two year old can clearly recognize simple desires and drives and how they motivate action, but this understanding is limited: "I want a cookie" consists of a target and a motivating attitude, and leads directly to goal-directed action: If I want a cookie, I see a cookie, and I'm allowed to have the cookie (Kushnir et al 2015), then my desire for the cookie *makes* me take the cookie. This is very different from the adult picture: I want a cookie, I see a cookie, I know I could have the cookie, but I may choose not to take the cookie (because I know I shouldn't eat too many cookies), or I may choose to get a cookie elsewhere (because there is a shop with better cookies on my route to work), or I may save the cookie for later (because I know I will want something sweet after lunch). Thus, to an adult, desires

constitute one input to a complex planning process over hypothetical representations, rather than an attitude toward an object that directly causes action. There are similar differences between 2 and 5 year old's understanding of awareness and perspective. The two year old can tell "whether X sees Y," but not "how Y appears to X." The two year old can tell, at least implicitly (Low & Watts 2013), that a parent who didn't see a toy being put on a shelf won't immediately search that shelf when prompted to retrieve the toy. But, when asked to explicitly describe a mistaken belief, or predict an action on the basis of that belief, children 3 and under tend to struggle. Thus, while the two year old understands "awareness" as a kind of direct epistemic access to the environment, older children seem to recognize beliefs as distinct *representations*, which track the environment in some way, but which are importantly fallible and may "misrepresent."

With this in mind, we can now articulate our main questions of interest. The first question is: why beliefs and desires? That is, why do we so consistently develop a folk psychology that explains behavior in terms of these two fundamental categories of mental states? Of course, "belief" and "desire" are both broadly defined commonsense concepts, so in order to address this question we will need to formally characterize the representational, relational, and functional features that distinguish these categories. The second question addresses the developmental trajectory from lean, minimally-representational ToM to richly representational ToM that we've described in this section. In particular, does this trajectory reflect the inductive behavior of a rational inference mechanism, and if so, what drives the inductive transitions from lean to rich? Obviously the "lean/rich" distinction is a graded, rather than binary concept, and we will need to characterize what it means to have a "leaner" or "richer" psychological model. Before presenting the formal framework we derive to answer these questions,

however, we shall review the philosophical background and motivation for our rational constructivist approach to ToM.

2.2 Conceptual and philosophical motivations

Now that we have laid out the empirical motivation for our questions of interest, we shall provide conceptual and philosophical motivation for the approach we intend to take. Our rationalist approach is largely motivated by the "Theory-theory" perspective. In this section we outline this motivating perspective, contrast it with other approaches to understanding cognitive development, and describe how we can transform this conceptual framework into a computational one.

2.2.1 Explaining cognitive development

Any attempt to understand the development of our Theory of Mind must address a recurring tension that underlies cognitive science in general. On the one hand, adults seem to possess abstract, coherently structured knowledge and representations, which allow us to interpret, predict, explain, and control our surrounding environment. But at the same time, this abstract knowledge is not obviously present in young children; somehow, we seem to acquire this knowledge from our concrete, noisy, often incomplete sensory evidence. How, then, could we extract these abstract, structured representations from sensory experience, especially given that our representations play a critical role in allowing us to interpret that experience in the first place? This tension is especially pronounced in understanding the development of ToM, which inherently involves abstract, unobservable entities.

Historically, attempts to resolve this tension are often grouped under one of two labels. Traditionally *nativist* accounts assert that the coherently structured knowledge and concepts we possess could not be learned from experience. Such accounts often appeal to a *poverty of the stimulus* argument (Chomsky 2006); essentially, that the coherent knowledge structures we possess (e.g. the grammar for speaking a given language) are underdetermined by the data we receive (e.g. examples of valid sentences and exchanges in that language). That is, the evidence we receive is, from a learning-theoretic perspective, insufficient to reliably infer the knowledge we possess from that evidence alone (Gold 1967). Thus, such accounts propose, much of the coherent, abstract structure apparent in our knowledge must be part of our "initial learning state," i.e. must be innate. In Chomksyan linguistics, this innate structure constitutes the Universal Grammar (Chomsky 1986); in more recent, general cognitive science accounts, they take the form of innate knowledge "modules" or core knowledge systems (e.g. Pinker 1997, Spelke & Kinzler 2007), some of which remain latent until later development.

Conversely, traditionally *empiricist* or *associationist* accounts suggest that our cognitive capacities reflect a collection of context-dependent "modules" or functions, acquired through extremely general learning mechanisms that extract statistical regularities between inputs and outputs. Recent empiricist accounts leverage the technical developments in neural networks (e.g. Elman et al 1996) or dynamic systems (e.g. Thelen & Smith 1994). Under these accounts, the abstract structures that appear to *underly* our cognitive development are, in fact, *emergent* from this collection of learned associations and functions.

In truth, it is somewhat misleading to regard any one account as "purely nativist" or

"purely empiricist," or to view "nativism" and "empiricism" as distinct, mutually exclusive theories of human cognition. Even core-knowledge theorists acknowledge that sufficient differences in available evidence influence the development of children's knowledge, at least somewhat (Spelke & Kinzler 2007). Furthermore, much of our knowledge can only be realized in an appropriate environment- consider, for example, an abstract understanding of how computers work, or learning to comprehend novel slang expressions that do not adhere to any compositional grammar. Conversely, any empiricist account must take *something* as given, something on which subsequent statistical learning can be founded. This often takes the form of a base set of perceptual primitives (e.g. color experiences and edge-detectors) and/or an innate set of statistical learning mechanisms.

Rather than viewing nativist and empiricist accounts as distinct and mutually exclusive theories of human cognition, we believe it is more informative to view nativism and empiricism as occupying different regions in a broader conceptual space, one with (at least) two relevant dimensions. The first dimension is whether, and to what degree, our knowledge is learned rather than built-in. The second is whether, and to what degree, our knowledge consists of abstract, richly structured, domain-general representations and mechanisms, or lots of distinct, separable modules, each of which serves a distinct context-specific function. As we develop our perspective, the relevant question is not whether our ToM-related knowledge is innate and abstract or learned and context-specific, but *what* aspects specifically are learned, what is innate; what skills could be explained as a context-specific instantiation of a domain-general mechanism, and what skills are difficult to explain without innate, domain-specific knowledge structures? With this in mind, our approach is motivated by the hypothesis that what is

innate are not complete, comprehensive abstract structures, but a set of primitive "conceptual building blocks" in which abstract, structured knowledge can be acquired and represented. This is sometimes referred to as a "minimal nativism" (e.g. Goodman et al 2011, Kemp et al 2007), which posits a small set of *constraints* on the learning process, rather than a wide range of pre-existing structured representations.

At a high-level, this perspective is closely related to Piaget's theory of constructivism (Piaget 1964, Wadsworth 1996), or the more recently named "Theory-theory" (Gopnik & Meltzoff 1997). Under a constructivist perspective, our abstract knowledge is represented in the form of intuitive theories, which are incrementally learned (or "constructed") from sensory experience. The Theory-theory makes the more specific assertion that these intuitive theories are, in fact, very much like "theories" in science, and that our learning and cognitive development proceed very much like theory-revision in scientific research. While Piaget's constructivist theory was a promising attempt at resolving this tension, it lacked any detailed account of how such learning processes might work. However, more recent developments in Machine Learning provide extra tools for modeling how such knowledge could be represented and acquired.

2.2.2 Intuitive theories and the Theory-Theory

In trying to explain human cognitive development as a process of intuitive theory-revision, Theory-theorists identify a number of key characteristics shared by our intuitive theories, which help distinguish a Theory-theory account from more nativist or empiricist accounts (Gopnik & Wellman 2012). First is the distinctive structure of our intuitive theories, which involve coherent, abstract representations of the world and its underlying causal mechanisms, and often include hidden, posited entities. Theories also

tend to be hierarchically structured, enabling prediction and explanation at multiple levels of abstraction or domain-specificity. The term "framework theory" is sometimes used to describe more abstract, domain-general theories, which specify the applicable entities and relations for a domain, rather than specifying the low-level details of those relations. A framework theory can also include some number of *over-hypotheses* (Kemp et al 2007) which constrain how the relevant predicates can be instantiated. Second, our theories serve distinctive cognitive functions, allowing us to perform a wide range of predictions and counterfactual inferences across a wide range of domains. Third, theories are dynamic, and change in light of new evidence. Importantly, children's intuitive theories can change in local, specific ways (e.g. learning to better predict an object's visible trajectory) *and* in broad, high-level principles (e.g. learning that objects continue to exist even when out of sight).

More recent developmental data reveal additional dynamic features of children's intuitive theories, in particular the important role of statistical information and probabilistic contingencies in children's learning. Even infants are sensitive to sampling distributions and can infer preferences or psychological intentions from statistical information (Xu & Denison 2009, Kushnir et al 2010). Second, informal experimentation and intervention play an important role in how children learn about causal structure. Third, children's theory change relies on variability, often testing and assessing multiple hypotheses simultaneously, and can generate new hypotheses by gradually "perturbing" existing hypotheses. This contrasts with a more traditional, learning-theoretic view of children's learning as a search process that points to a single hypotheses out of a pre-defined space of possible alternatives.

While there is much debate over the specific details of what concepts children acquire

when (and in what order), the data show a clear developmental picture in children's learning, characterized by the common features described above. Much of this data comes from recent studies motivated in large part by this Theory-theory perspective, and indeed the Theory-theory has been quite fruitful in directing new research. However, while the data generated by these experiments are rich and informative, the Theory-theory has, until recently, lacked a plausible computational account of the learning mechanisms involved in high-level, abstract theory revision. Thus, while the data paint a vivid picture, the constructivist project suffered from theoretical vagueness.

In the past decade, however, there has been a resurgence of interest in constructivist accounts, motivated by the development of new formal machinery. Several authors have proposed that Causal Graphical Models (CGMs) and Hierarchical Bayesian Models (HBMs) can provide a computationally-grounded account of how we represent, acquire, and use intuitive theories (Gopnik & Wellman 2012), as well as the process through which we revise these intuitive theories (Henderson et al 2010). We now present a computational framework that applies these concepts to human ToM.

3 Computational framework

3.1 High level overview

Here we provide a high level overview of our computational framework. At its core, the models in this framework represent a single "observer" agent making inferences and predictions about the behavior of one or more other agents (actors). In order to properly define and evaluate our questions of interest, we will need to represent inference

problems (and solutions to those problems) at three levels of abstraction. We provide an intuitive overview of these three levels before presenting a hierarchical framework that incorporates all three.

3.1.1 Level 1

A Level 1 problem corresponds to inferring a "psychological explanation" (formalized as a kind of probabilistic generative model) for a particular episode of a particular agent's behavior, and then manipulating this explanation to reason about the episode. This reasoning can include:[6]

- Behavior prediction; e.g. "What will the actor do next, given what they've done so far?"
- Psychological inference; e.g. "What does the actor want/believe, given the way they behave?"
- Control inference; e.g. "How do I get the actor to do X?"
- Counterfactual inference; e.g. "What *would* the actor have done if this object weren't here?"

Importantly, the observer will often have to do both of these tasks (infer a psychological explanation, *and* manipulate that explanation to do reasoning) contemporaneously and dynamically as the episode elapses. After all, an important feature of our ToM is that we

[6]Note that the observer's ability to perform these inferences depends on the structure and content of their explanation; e.g. an observer who does not attribute belief states to actors will obviously not be able to infer an actor's belief

use it to navigate social environments. It is therefore important that our framework captures not only how the observer explains a completed episode, but how the observer progressively updates their explanation as they acquire more evidence.

3.1.2 Level 2

The inputs to a Level 1 inference problem are a (partial) episode of an actor's behavior (the data), and a context-general "model" of that actor's behavior generating process (which will often include hidden psychological states posited by the observer).[7] A Level 2 problem corresponds to inferring such a model for a particular actor, given multiple episodes of that actor's behavior (usually across multiple different environments) and a higher level "model template" or "framework theory" (Level 3 inference) that constrains the space of possible models. Inferences at this level may include

- Inferring an actor's preferences or values over outcomes
- Identifying actor-specific behavioral quirks (e.g. a frequent behavioral pattern that serves no apparent purpose)
- Identifying actor-specific cognitive constraints (e.g. that an actor has limited vision/visual range, or that an actor has/lacks the ability to form complex plans)

A Level 2 problem may involve learning an actor model from a large number of episodes, or progressively updating an existing model on a sequence of episodes. It may involve

[7]We put "model" in quotes because technically, this object does not specify a single model, but a comprehensive set of instructions for constructing a model in a particular context

updating a single parameter or constraint in the model, or constructing a new model from scratch for a previously unknown actor. In either case, the inputs to a Level 2 problem are the data, which consist of multiple episodes of a single actor's behavior, and the model template, which is inferred at Level 3.

3.1.3 Level 3

Intuitively, a Level 3 problem corresponds to learning a "psychological framework theory," which specifies

1. an ontology of mental entities (and their corresponding domains/types) that can be used to construct a psychological model,
2. a set of constraints on how these entities may interact with external stimuli, observable behavior, and other mental entities,
3. a set of constraints on the functional and algorithmic forms of relations between these entities, and
4. a probability distribution over possible configurations of entities and relations, subject to constraints.

A single configuration of each (i.e. a set of allowable entity types, a set of allowable relations between those entity types, and a set of allowable functional and algorithmic forms mediating those relations) provide the "model template" for the actor-specific models inferred at Level 2, and these actor-specific models provide a template for the episode-specific explanations inferred at Level 1. The observer's "psychological framework theory" specifies both the space of possible templates, and a prior distribution

$P(t)$, over templates, where $P(t)$ captures the prior probability that the behavior of a (previously unknown) actor will be explained by an actor model with model template t.

The data for a Level 3 inference problem comprise multiple episodes of multiple different actors' behavior, ideally across multiple different systems, and can be performed contemporaneously with Level 2 and Level 1 inference. Like Level 2 inference, Level 3 problems may involve constructing a full theory from scratch over a large body of data, or updating one or more constraints in an existing theory from a small number of new observations. Intuitively, these updates involve inferring general psychological properties about actor behavior; e.g. inferring that actors are (usually) constrained by their visual awareness, or that actors tend to have (peaked and sparse) or (flat and dense) preferences over a fixed set of possible outcomes, or that younger actors are less likely to engage in complex planning than older actors.[8]

3.2 Technical background

Here we review the formal machinery we shall use to construct the framework.

3.2.1 Bayesian inference and Bayesian models of cognition

The basic claim underlying the Bayesian approach to cognition is that much of our cognitive behavior can be accounted for as approximately rational probabilistic inference. At the core of any Bayesian theory of cognition is an observer model: we represent a learning or inference agent as a rational observer with some prior knowledge or

[8]Though the demonstrations in this thesis won't involve any observable demographic features like age

expectations, and ask how that agent would optimally respond to a given stimulus or event. We can use this to provide theoretical justification for the hypothesis that a particular aspect of human knowledge or cognition is "learned" (by demonstrating how it *could* be learned). We can also invert this analysis to ask what prior knowledge, representations, or expectations lead to behavior consistent with what we observe in humans.

An observer model consists of two components. The first is the observer's hypothesis space, $\mathcal{H}$: this represents the possible explanations that the observer may consider for a given class of stimuli. At an abstract level, a hypothesis induces a probability distribution over possible observations,[9] which allows the observer to perform probabilistic inference and predictions about a given class of stimuli. If, for example, the stimuli in question consist of two observable features $s = (f_1, f_2)$, then a hypothesis in this domain induces a joint probability distribution $P(f_1, f_2)$. Under this hypothesis, the observer can compute the likelihood of a particular observation (i.e. stimulus with certain feature values), and can also predict or infer the value of one feature by observing the other.

The second component is a prior distribution over hypotheses, $P(\mathcal{H})$. This represents the observer's prior degree of belief in each hypothesis in the absence of any relevant evidence. When presented with some evidence E (i.e. one or more observations), the observer updates the degree to which they believe each hypothesis h according to Bayes' Theorem:

[9]Or, if the hypothesis includes posited hidden states, a joint distribution over configurations of the hidden states *and* possible observations

$$P(h|E) = \frac{P(E|h)P(h)}{P(E)} \tag{3.1}$$

Here, $P(h|E)$ is the *posterior probability* of h given E: this is the degree to which a rational observer would believe hypothesis h given evidence E and prior beliefs $P(h)$. The denominator $P(E)$ is a normalizing term which we may ignore for the scope of this dissertation,[10] instead writing Bayes' Theorem as $P(h|E) \propto P(E|h)P(h)$.

Most often, the observer's hypothesis space is defined in terms of a generative model, which specifies a probabilistic process for generating (simulated) observations, and induces an implicit probability distribution over possible observations (in section 3.2.4, we shall illustrate how to make this implicit distribution an explicit object). A generative model is defined in terms of its structure and parameter space. The structure comprises the set $\mathcal{X}$ of variables in the model (including observable variables and any hidden variables posited by the observer), and the dependence relations between these variables, encoded by a dependency relation R. This dependency relation determines an efficient way to parameterize the joint probability distribution induced by the model. In particular, if we let $\mathcal{X}$ denote the set of variables in a generative model, and $par(x)$ denote the set of variables on which x is dependent (i.e. all $y \in \mathcal{X}$ such that $y \to x$), then the joint distribution $P(\mathcal{X})$ factors as[11]

[10]We can ignore the denominator because our framework only requires computing a ratio of posteriors $P(h_1|E)/P(h_2|E)$, in which case the normalizing terms cancel out

[11]In general, the *Markov Property* for directed acyclic graphs (DAGs) entails this correspondence between the dependency structure of the graph and the factorized form of the induced joint distribution

$$P(\mathcal{X}) = \prod_{x \in \mathcal{X}} P(x|par(x)) \tag{3.2}$$

Here, each term $P(x|par(x))$ corresponds to a vector of parameters which determine the probability of x taking on a value in its domain, given the values of its parent variables. The set of all terms $P(x|par(x))$ constitute the parameter space Θ for the generative model. Thus, the structural part of a generative model (i.e. its variable set and dependency relation) determine a hypothesis space of parameter vectors, and each parameter vector corresponds to a single hypothesis (i.e. distribution).

3.2.2 Hierarchical Bayesian Models

In many cases, we wish to model inference at multiple levels of abstraction, across different scopes of data. For example, suppose we wish to infer the bias w of a particular coin after observing some sequence of flips $E = (x_1, x_2, \ldots, x_n)$. This process has a simple generative model (see figure 3.2.1): each flip is generated by a Bernoulli distribution with weight w, and a sequence of n flips is generated by n independent Bernoulli samples. We

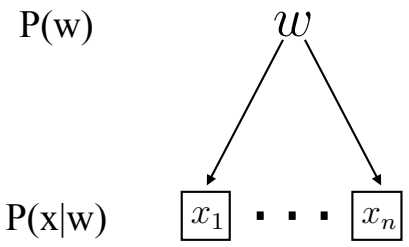


Figure 3.2.1: Generative model for a sequence of n coin flips with bias w

can use apply Bayes' rule as described above to infer the bias of this particular coin:

$$P(w|x_1, \ldots, x_n) \propto P(x_1, \ldots, x_n|w)P(w) = \prod_{i=1}^{n} P(x_i|w)P(w)$$

The observer's relevant background knowledge for this inference consists of the functional form for $P(x_i|w)$ (i.e. the probability of observing outcome x_i, given that the true weight of the coin is w), and a prior distribution $P(w)$ over possible weights of the coin. If the observer has no relevant past experience, the prior $P(w)$ may be uniform over $(0, 1)$, indicating that the observer has no reason to believe that one range of weights will be more likely than any other range of equal length. However, even if we have no past experience with this particular coin, we may suppose that our past experience with other coins might provide relevant evidence for experience with this new coin. For example, if most coins that we have encountered in the past are approximately fair, we might expect this new coin to be approximately fair. In order to represent this sort of generalization from other relevant experience, we need to add another level of inference to the model. Suppose that we now get to observe sequences of coin flips for several different coins, all

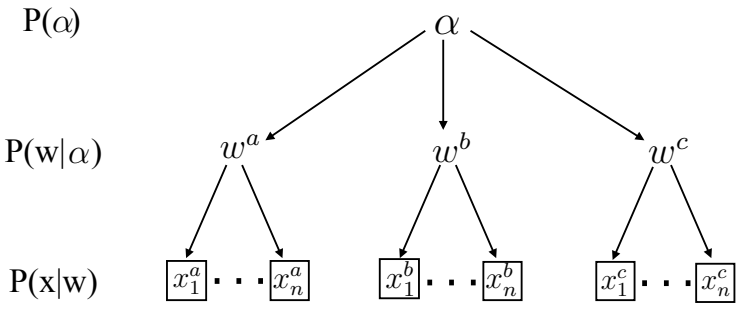


Figure 3.2.2: Generative model for three sequences of n coin flips for each of 3 coins

minted in the same batch at the same factory, and therefore (we shall assume) equally likely to bear defects that affect the bias. We can generalize the generative model in figure 3.2.1 to a hierarchical model with an additional level (see figure 3.2.2). According to this model, the prior distribution over w is itself determined by a *hyperparameter* α (i.e. $P(w|\alpha)$), and the bias for each of the coins is drawn from this same prior. This allows the observer to draw on past experience with different coins to update their estimate of α, which in turn allows for more efficient and accurate estimates of the new coin's bias. Specifically, if we let E_a, E_b, and E_c denote the evidence (sequence of flips) for coins a, b, and c respectively, the observer can apply Bayes' Rule to compute

$$P(\alpha|E_1, E_2, E_3) \propto P(E_1, E_2, E_3|\alpha)P(\alpha)$$

According to the observer's generative model, the sequences E_1, E_2, and E_3 are conditionally independent given α, so the term $P(E_1, E_2, E_3|\alpha)$ decomposes into terms of the form $P(x_j^i|\alpha)$. For each of these terms, the observer can marginalize out the bias parameter w^i, by computing the integral

$$P(x_j^i|\alpha) = \int P(x_j^i|w^i)P(w^i|\alpha)dw^i$$

This illustrates the basic principles of Hierarchical Bayesian Models: each lower level of inference is constrained by prior knowledge at higher levels (e.g. inference about the bias of a coin is constrained by the higher level parameter α, inference about a particular coin is constrained by its bias parameter), and prior knowledge at higher levels can be inferred and updated from evidence collected at a wider scope (e.g. coin flips of multiple

coins from the same factory, versus coin flips from a single coin). We shall draw on these principles in order to generalize our framework to the three levels of abstraction.

3.2.3 Bayesian Theory of Mind

Bayesian modeling has been used to account for a wide variety of cognitive functions, including object perception (Kersten & Yuille 2003), prediction of object trajectories (Weiss & Adelson 1998), visual feature inference (Griffiths & Austerweil 2009), and word learning (Xu & Tennebaum 2007). More recently, Bayesian models have been applied to mental inference and ToM tasks (e.g. Baker et al 2009, Baker et al 2011, Hamlin et al 2013). These models use the same three basic components described above: a hypothesis space of generative models, a prior distribution over hypothesis, and the application of Bayes' Theorem.

To quickly illustrate the core components of Bayesian ToM, consider a simple example in which actors navigate a two-dimensional grid world. Figure 3.2.3 illustrates a simple generative model that an observer might posit to explain an actor's behavior in such an environment.

According to the generative process posited by this observer, the actor's behavior at time t is determined by the actor's goal state g, the layout S of the current grid environment, and the actor's physical state (i.e. location) at time t. The actor's *goal distribution* is determined by the set of feasible goals in the current grid environment, and a bias parameter b specific to the actor. The observer can use this model to infer the actor's current goal, given a partial observation of the actor's behavior. Suppose, for example, that we observe the actor's first action a_0 (in addition to the grid layout S and the actor's initial location x_0). The observer can infer a posterior distribution over the

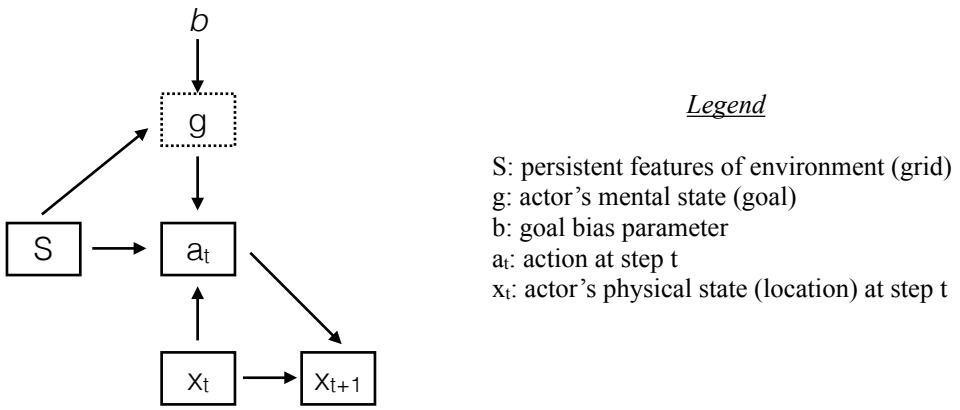


Figure 3.2.3: Simplified "rational planning model" (Baker et al 2009). Solid borders denote observable variables, dotted borders denote posited hidden states, and no borders denote parameter variables. We use dynamic graph notation to save space

actor's current goal by applying Bayes' Rule

$$P(g|S, x_0, a_0) \propto P(a_0|S, x_0, g)P(g)$$

The first term $P(a_0|S, x_0, g)$ denotes the probability that the actor would take action a_0 in that initial state, given that the actor's goal is equal to g. This term appears in the Markov Factorization of the model in figure 3.2.3, and therefore corresponds to a parameter table in the observer's generative model. In this case, we use a simplified "rational planning model" (Baker et al 2009) in which the actor follows a shortest path to the goal state (up to some probability of error). The second term, $P(g)$, corresponds to the bias parameter b. Thus, an observer with a fully parameterized generative model can compute the probability that the actor has a particular goal in mind when choosing action a_0 from initial state $S_0 = (S, x_0)$. The observer can also use this model to directly

predict future behavior by *marginalizing out* the psychological states, according to the equation

$$P(a_1|S_0, a_0) = \sum_g P(a_1|S_0, a_0, g)P(g|S_0, a_0)$$

where $P(g|S_0, a_0)$ is the posterior probability from the previous equation.

This illustrates the basic components of the Bayesian ToM (BToM) approach: a parameterized generative model for simulating the actor's behavior, which can be manipulated via Bayesian inference to reason about an actor's behavior and psychological states. We will draw heavily on the BToM approach to construct Level 1 of our framework. We then draw on the principles of Hierarchical Bayesian Models to generalize this for inference at higher levels of abstraction.

3.2.4 Probabilistic Programs

Thus far, our discussion has mostly been framed at the computational level of analysis. That is, we have discussed the kinds of problems that our observer will need to solve (i.e. the structure of the data and inferences), but not the actual procedures or algorithms that the observer will use to solve these problems. Much of the literature on Bayesian cognitive modeling is focused on the computational level, though as we explain in the next section, some of our questions of interest will depend on interactions between algorithmic-level constraints (e.g. the observer's representational resources and their capacity to manipulate those representations) and computational-level performance. For this reason, we define our framework using the formal machinery of functional probabilistic programming languages (PPLs), which are useful for exposing these sorts of interactions between different levels of analysis, and have been proposed as a natural way

to model "reasoning about reasoning" (Stuhlmuller & Goodman 2016). Here we provide a brief primer on three core components of PPLs which are conceptually important for our framework.[12]

The first core component is a set of stochastic primitive functions. For example, the stochastic primitive function $flip(w)$ returns a 1 with probability w or a 0 with probability $1 - w$. We can compose stochastic primitives to define more complex probabilistic programs. For example, the program below simulates n coin flips with weight w, then returns the total number of successes.

```
var nFlips =function(n, w){
    var outcomes = repeat(n, flip(w))
    return sum(outcomes)
}
```

This program constitutes a procedure for generating samples from a probability distribution, parameterized by its inputs (in this case, a $Binomial(n, w)$ distribution). Alternatively, we can view the program as inducing a mapping from inputs to *distributions over* outputs, although this distribution is only implicit in the return values of this program. However, we can transform a procedure for generating samples from a distribution into an explicit *distribution object* using the second core component: a *marginalization* operator. In the language we use (WebPPL), this is the $Infer$ operator. Consider the following modification to the previous program:

[12]See appendix A for a more detailed explanation of PPLs

```
var nFlipsDist =function(n, w){
    return Infer(function(){
        var outcomes = repeat(n, flip(w))
        return sum(outcomes)
    })
}
```

A call to the first program- $nFlips(n, w)$- returns a sample drawn from a $Binomial(n, w)$ distribution, whereas a call to the second program- $nFlipsDist(n, w)$-returns a $Binomial(n, w)$ distribution object, which we can then query with built-in functions; e.g. if we define $Dist = nFlipsDist(n, w)$, we can use $sample(Dist)$ to draw a sample from this distribution, or $Dist.score(x)$ to compute the log-likelihood of observing x under this distribution.

The final core component is an operator for conditioning on observations: WebPPL provides several such operators, the most straightforward of which is $Condition(A)$, where A is boolean. Including a line of this form inside the scope of an $Infer$ statement[13] conditions the resulting distribution on the observation that proposition A is true. For example, consider the following modification to $nFlipsDist$:

```
var nFlipsDist =function(n, w, obsNum, obsVal){
    return Infer(function(){
        var outcomes = repeat(n, flip(w))
        Condition(outcomes[obsNum] == obsVal)
        return sum(outcomes)
}
```

This function requires two additional inputs: $obsNum$ (an integer $< n$), and $obsVal$

[13]Condition statements have no effect outside the scope of an Infer statement

(boolean). The line Condition($outcomes[obsNum] == obsVal$) adds the condition that flip $obsNum$ resulted in outcome $obsVal$. Intuitively, we can understand how this program works in terms of rejection sampling:[14] The Infer operator repeatedly runs the generative model inside it, and rejects any sample which violates the statement in Condition. It then computes the empirical distribution of the requested value within the population of accepted samples. Thus, a call to $nFlipsDist(10, .5, 1, H)$ returns a distribution over total number of successes in 10 flips, given that the first flip was a success.

These three core components- stochastic primitives, a marginalization operator, and a conditioning operator- will allow us to compactly define complex hierarchical models and generalized procedures for inference over those models.

3.3 Level 1: details

3.3.1 Level 1 data and inference tasks

Intuitively, Level 1 corresponds to reasoning about a particular instance of a particular actor's behavior. To represent the data involved in Level 1 reasoning, we shall use Markov Decision Process (MDP) notation. Formally, each episode takes place in an *MDP system* $\{S, states\_toActs, T)$, where

1. S is the set of possible *system states*. We use s_t to denote the set of all relevant observable features of the system at time t. For detailed analysis, it may be useful to further demarcate s_t into individual feature variables (e.g. f_t^i for the value of the

[14]Though rejection sampling is generally intractable, WebPPL includes several more efficient inference methods. See Appendix A for more details

ith feature at time t), or to distinguish between the observable *environment state* (e.g. layout of the room) and the observable *actor state* (e.g. the actor's current location in the room). For this explanation, however, we use s_t to denote a complete description of the (observable) system-state at time t.

2. $states\_toActs : S \to \mathcal{P}(A)$ is a function mapping a system state s to the set of possible *actions* that are feasible from that state.

3. $T : S \times A \to S$ is a *state transition function*, i.e. $T(s, a)$ is the new state resulting from taking action a in state s.[15]

For now, we shall assume that the observer has full knowledge of S, $states\_toActs$, and T, which we can think of as encapsulating the "intuitive physics" of the system. To help illustrate our framework, we provide a simple two-dimensional "gridworld" environment that we refer to throughout this section. An example system state for such an environment is shown in figure 3.3.1:

In this case, the system state s_t consists of three features: the grid layout of the environment (i.e. locations of any objects and features), the location of the actor, and the actor's "inventory," which contains any objects the actor is currently holding. For this example, we define $A = \{$'up', 'down', 'left', 'right', 'pickup'$\}$: the first four actions denote movement by one cell in the corresponding direction, and the 'pickup' action adds any objects in the actor's current location to the actor's inventory.

[15]Generally, the state transition function may be probabilistic, in which case $T(s, a)$ is a distribution over possible next states. We will only consider systems with deterministic transition functions for now

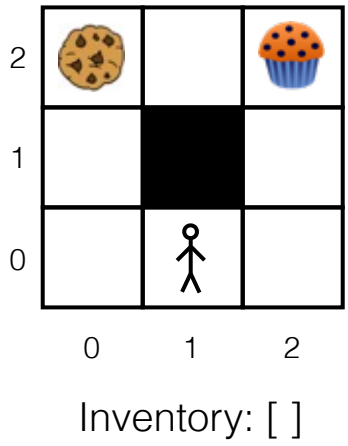


Figure 3.3.1: Toy MDP system which we shall use to illustrate the core components of our formal framework. Black squares denote walls, which cannot be traversed or occupied

The data for a Level 1 problem comprise a (possibly incomplete) observation of a single episode. We use $d = \{obs_1, \ldots, obs_n\}$ to denote a trial observation, where each $obs_i = (var_t^i, val_t^i)$ consists of an observable feature variable var_t^i (either a state variable or action variable), and an observed value for that variable. An example of a trial observation for the example system above is shown in figure 3.3.2. Using the notation just defined, we can represent this trial observation as

$$d = \{(state_0, s_0), (act_0, \text{ 'left'}), (state_1, s_1), \ldots\}$$

Now that we have characterized the data involved in Level 1 problems, we can define the inference tasks. Intuitively, a Level 1 problem involves two tasks, which are usually solved contemporaneously and dynamically as the episode elapses. The first task is to infer a "trial explanation" of the data, which we define in the next section. The second task is to manipulate this explanation to reason about the ongoing episode. Before

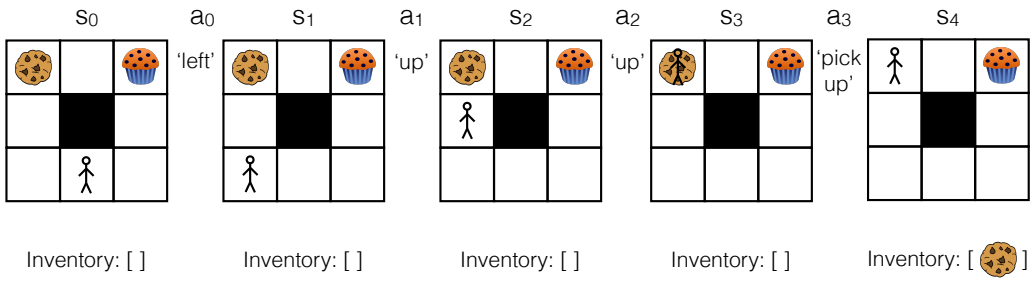


Figure 3.3.2: Example of a full trial observation in a grid world system

providing a formal characterization of the "trial explanation," we provide an intuitive explanation of what sorts of manipulations the observer will need to perform.

To this end, we distinguish between four kinds of Level 1 reasoning:

1. Action prediction: for example, if we restrict the data to the first state-action-state sequence of the episode shown in figure 5, the observer might need to predict the actor's next action, given their behavior so far.

2. Mental inference: this involves inferring the values of hidden psychological states attributed to the actor under the observer's psychological explanation. For example, if the observer's explanation includes a notion of rational goal-seeking behavior, the observer may need to infer the actor's goal from a few steps of the actor's behavior. Note that the structure and content of the observer's explanation constrains which mental inferences the observer can perform, e.g. an observer who does not attribute beliefs to actors obviously cannot infer beliefs.

3. Control reasoning: this is only applicable in cases where the observer can intervene on the system (either the environment or the actor), and the observer must determine a sequence of interventions for inducing some target behavior in the

actor. For example, if the observer can remove or relocate one of the two "treats" shown in figure 4, the observer may reason about how to control the actor's movement through a sequence of such interventions.

4. Counterfactual reasoning: this can encompass any of the previous types of reasoning, and involves the same mathematical operations, but is performed on "counterfactual" data. For example, the observer might wonder what the actor's first move *would have been* if the treat in the top left corner *had been* an apple instead of a cookie.

We define the observer's trial explanation as a type of probabilistic generative model which can be manipulated to perform Level 1 reasoning. A *psychological explanation* is a type of trial explanation with certain distinguishing features, which we expand on in the next section. For this paper, we shall focus on the first two types of reasoning for Level 1, though we can generalize this to counterfactual and control reasoning with minor augmentations to the model definitions.

3.3.2 Level 1: Inferring and reasoning with psychological explanations

Here we provide a formal definition of "trial explanation," illustrate how they can be manipulated to reason about an ongoing episode, and illustrate how an observer can both infer and manipulate this explanation contemporaneously using functional probabilistic programs.

To this end, let $d = \{obs_1, \ldots, obs_n\}$ be a (possibly incomplete) trial observation for some MDP system $\{S, states\_toActs, T)$. Let $\mathcal{V}$ be the set of *trial variables* occurring in

this observation, i.e.

$$\mathcal{V} = \{var_t^i | (var_t^i, val_t^i) \in d \text{ for some value of } val_t^i\}$$

Let D denote the set of all possible *completions* of d attainable in this system; i.e. the set of all *complete* trials consistent with the observations in d (if d is a complete trial observation, then D will be the singleton set containing d). A *trial explanation of* d is a type of probabilistic generative model, consisting of:

1. A variable set $\mathcal{X}$ which contains $\mathcal{V}$. We use $\mathcal{U}$ to denote $\mathcal{X} \backslash \mathcal{V}$. If $\mathcal{U} = \emptyset$, the explanation is "non-psychological"[16]

2. A binary dependence relation $R \subset \mathcal{X} \times \mathcal{X}$. For a variable $x \in \mathcal{X}$, we use $Par(x)$ to denote the *parents* of $\mathcal{X}$, i.e. $Par(x) = \{y \in \mathcal{X} | R(y, x)\}$.

3. For each variable $x \in \mathcal{X}$, a probabilistic program $f_x(Par(x); \theta_x)$ with inputs in $Par(x)$ and parameter vector θ_x.

Together, these components define a joint probability distribution which factors according to the equation below:

$$P(\mathcal{X}|d) = \prod_{x \in \mathcal{X}} P(x|Par(x), d) \tag{3.3}$$

Note that, by definition, the distribution $P(\mathcal{X}|d)$ only has support on configurations of $\mathcal{X}$ in which all observed variables have values consistent with those in d. If d is a

[16]though the converse is not true: as we explain in section 3.5.3, $\mathcal{U} \neq \emptyset$ does not necessarily imply that the explanation is psychological

complete trial observation, we may write this as $P(\mathcal{U}|d)$, to reflect that the relevant set of possibilities only varies over values of unobserved variables (i.e. those posited by the observer). If d is an incomplete trial observation, we instead write $P(\mathcal{U}, D)$ to denote that the relevant set of possibilities varies over possible completions of the trial *and* values of unobserved variables.

Before we explain this in more detail, it is important to note that equation 3.3 is the standard Markov factorization for causal graphical models. Indeed, a trial explanation can be interpreted as a type of augmented Causal Graphical Model (CGM), which we refer to as a Causal Process Model (CPM). The distinguishing feature of a CPM is that the functional and probabilistic relations between variables specify additional information at the algorithmic level of analysis, beyond the computational level. That is, a CGM specifies the causal relations between variables, and the conditional probabilities $P(x|Par(x), \theta_x)$, but is agnostic about the underlying processes or mechanisms that give rise to these probabilities. A CPM specifies additional, algorithmic-level information about a *procedure* generating samples from the distribution $P(x|Par(x), \theta_x)$. As we explain more thoroughly in Chapter 5, this extra level of information will be highly relevant for addressing our questions of interest.[17]

Intuitively, the induced distribution corresponds to the "computational-level"

[17]Also noteworthy is that the algorithmic-level information in a CPM can be "marginalized out" to obtain the corresponding CGM. This marginalization does not affect the probabilities. Conversely, each CGM can be associated with a "default" CPM by simply defining each program to draw samples from the appropriate categorical distribution. Thus, each CPM corresponds to a single CGM, but a single CGM may be realized via many CPMs

interpretation of the observer's trial explanation, while the model itself specifies additional algorithmic-level properties. That is, the factorized distribution captures the relational and functional structure among variables, while the model specifies the processes that mediate these relations. We can encode this model as a single program $f(queries)$, where $queries$ consists of any variable names in $\mathcal{X}$, and a call to $f(queries)$ returns a single sample from the distribution $P(queries|d)$. To better illustrate this construction, and how it can be used to perform Level 1 reasoning, consider the partial trial observation shown in figure 3.3.3.

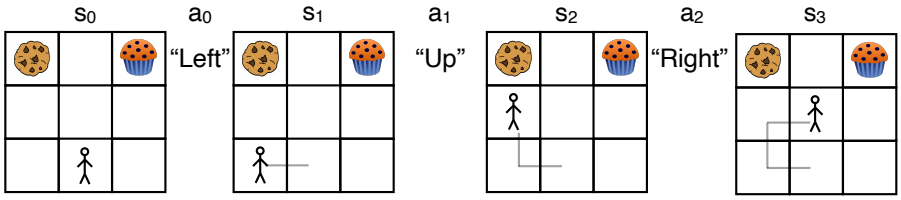


Figure 3.3.3: Partial trial observation. We assume that the actor cannot backtrack to previously occupied locations. Faded lines indicate the actor's path thus far

For this example, we shall assume that the system prevents "backtracking" (i.e. the actor may not move to the same location they were in previously), that the trial ends once the actor reaches and picks up either the muffin or the cookie, and that actors will not "back themselves into a corner" (i.e. will not enter a state that leaves them with no available actions, unless it is a terminal state). Under these assumptions, it is straightforward to compute the set of possible completions of this partial trial (figure 3.3.4).

Figures 3.3.5 and 3.3.6 illustrate two possible trial explanations for this partial

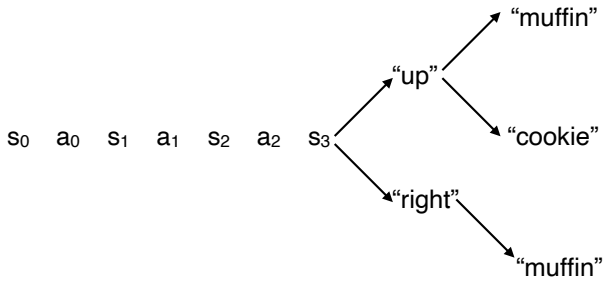


Figure 3.3.4: Possible completions of the partial trial in figure 3.3.3, consistent with system constraints

trial.[18] According to the explanation in figure 3.3.5, the actor's movement at time t is determined by the current system state s_t, a binary goal state which is persistent throughout the episode, and an error parameter ϵ. At each step, the actor follows a shortest path from their current location to the goal object (0 ='cookie', 1='muffin') with probability $1 - \epsilon$, but has an ϵ chance of making an error (choosing another move randomly). The actor's preference between cookies and muffins is captured by the preference parameter β: for a fixed value of β, the actor's goal is drawn at the start of the episode from a $Bernoulli(\beta)$ distribution. The explanation in figure 3.3.6 is similar, with two important differences. First, under this explanation, the actor will always follow a shortest path toward the current goal object (i.e. $\epsilon = 0$). Second, the goal variable is not fixed throughout the episode, but has chance ν to be randomly re-sampled in each step, from the same prior parameterized by β. Intuitively, the first model encodes the explanation that the actor has the same goal object in mind throughout the episode, but makes a "mistake" en route to that object, while the second model encodes the explanation that the actor never makes a mistake (given their current goal), but changes their goal mid-trial.

[18]We use a condensed, intuitive psuedocode to illustrate programs in these figures

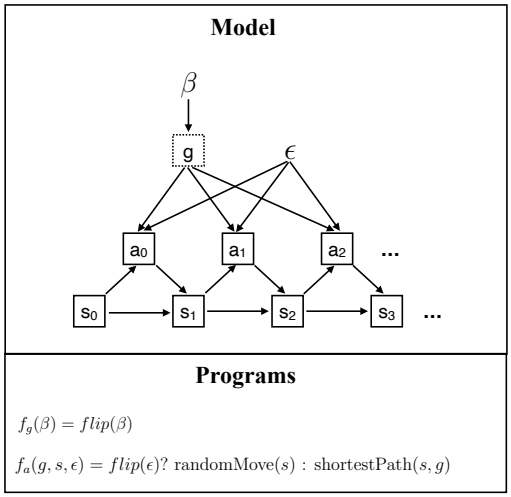


Figure 3.3.5: An "error-prone fixed-goal" explanation of the partial trial in figure 3.3.3. Note that the syntax $A?B:C$ corresponds to "if A then B otherwise C"

Importantly, a trial explanation is not a static description of "what happens" in the trial, but a dynamic model that can be manipulated to reason, via simulation, about the actor's past, present, possible future, and counterfactually possible behavior. This is closely related to the idea that human beings can reason about counterfactuals and causal relations by "mentally simulating" hidden or counterfactual events according to a generative model (Gerstenberg et al 2014).

To illustrate how this manipulation works, consider an observer who infers the first (error-prone fixed-goal) explanation. Under this interpretation, the actor must have made a mistake, either in the first move (if $g = Cookie$) or the third (if $g = Muffin$). The observer can manipulate their explanation to estimate the probability of each possibility. On a computational level, this requires manipulating the joint probability

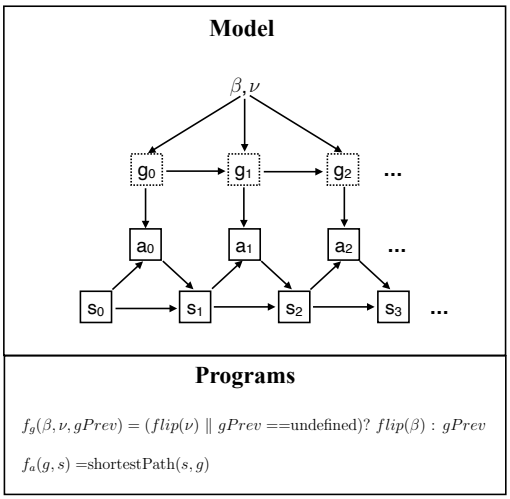


Figure 3.3.6: A "deterministic-action mind-change" explanation of the partial trial in figure 3.3.3

distribution induced by the explanation using Bayes' Rule, i.e.

$$P(g = C|d) \propto P(d|g = C)P(g = C) = \left(\prod_{obs \in d} P(obs|g = C) \right) \beta$$

Note that $P(g = C)$ is equal to parameter β (which is learned via Level 2 inference), and each term $P(obs|g = C)$ is equal to the distribution over return values induced by f_a. Similarly, the observer can predict the actor's next move by marginalizing out the goal states, i.e.

$$P(a_3|d) = P(a_3|d, g = C)P(g = C|d) + P(a_3|d, g = M)P(g = M|d)$$

where the posterior goal probabilities $P(g|d)$ are computed according to the previous equation.

On an algorithmic level, the observer's explanation is defined in terms of an "actor

model," which is inferred via Level 2 inference. We shall explain the actor model in greater detail in the next section, but for now, we can define it as a recursive loop for generating simulated trials, where each iteration executes the following three programs (in order)

1. $uf(s, m, \theta_u)$ takes the current system-state s, the actor's current mental state m, and a parameter vector θ_u, and returns an *updated* mental state m'

2. $af(s, m, \theta_a)$ takes a current system-state s, the actor's current mental state m, and a parameter vector θ_a and returns an action a

3. $T(s, a)$ computes the next system state via the system's transition function. If this is a terminal state (defined by either the system constraints[19] or by the actor's goal state), then the recursion halts, and the function returns a fully specified trial (including values for all hidden variables in the model). Otherwise, the recursive function is applied to the next state.

If we let $actorModel()$ denote the functional PP that executes this recursive loop and outputs a complete trial, we can use the PPL's Infer and Condition operators to compute the distribution $P(query|d, actorModel)$, where $query$ is any set of variables not observed in the data. The general form for this is[20]

[19]E.g., most systems will have a "maximum number of actions" cap after which the trial ends

[20]While this form is theoretically sound and intuitively clear, it is generally intractable outside of simple cases. See Appendix B for full specification of the actual programs

```
var Level1_Inference =function(data, query, actorModel){
    return Infer(function(){
        var trial = actorModel()
        Condition(checkData(trial, data))
        return getQuery(query, trial)
    })
}
```

Intuitively, the code inside the Infer operator first generates a simulated trial by running the $actorModel$ program. The helper function $checkData(trial, data)$ returns 'true' if the trial is consistent with the observations in $data$, and 'false' otherwise. The Condition statement therefore conditions the distribution on the observed data. Finally, the helper function $getQuery$ extracts the values of all variables requested in $query$ (recall that the trial returned by $actorModel()$ includes values for posited hidden variables, so $query$ may include requests for the actor's mental states). Thus, a call to this program returns an object encoding the distribution $P(query|data, actorModel)$. Furthermore, this general form allows the observer to update their explanation dynamically as new observations arrive, by adding new observations to the input for the $data$ argument. We illustrate this dynamic updating in figure 3.3.7, using the partial trial shown in figure 3.3.3 and an "error-prone fixed-goal" actor model.

To summarize: for a fixed trial observation d and actor model A, the observer's *trial explanation of* d can be characterized at both the computational and algorithmic levels of analysis. At the algorithmic level, the explanation is the program $f(queries) = Level1\_Inference(d, queries, A)$. At the computational level, the explanation is the mapping from queries to probability distributions induced by this program (alternatively, we can define it as the full joint probability distribution

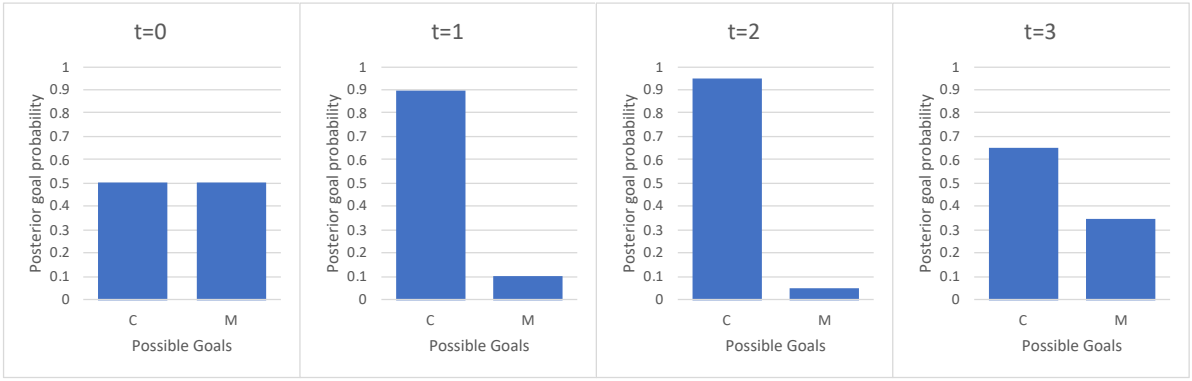


Figure 3.3.7: A illustration of how the observer dynamically updates their explanation as an episode elapses. Each chart depicts the posterior distribution over the actor's goal (according to the observer's actor model) after observing the corresponding trial step. The first chart depicts the prior distribution, as it is conditioned on no action observations. The next two observations significantly increase the posterior probability that the actor is targeting the cookie. After the next observation, however, this posterior probability of $g = C$ sharply decreases (to reflect that the actor may have been targeting the muffin all along, but made a mistake in their first action) and the posterior probability of $g = M$ increases

$P(\mathcal{U}, D|d)$, from which we can obtain the query distributions by marginalizing out non-queried variables).

3.4 Level 2: details

3.4.1 Actor models

A Level 2 problem involves inferring the actor-specific template which constrains Level 1 reasoning. Intuitively, an actor model captures the (observer's hypothesis about the) actor's values, preferences, cognitive constraints, and behavioral dispositions in an abstract representation (i.e. without reference to trial- or system-specific features). Formally, an actor model $\mathcal{M} = \{H, af, uf, \theta\}$ consists of

1. A set of "mental variables" $H = \{(h_1, type(h_1)), \ldots, (h_n, type(h_1))\}$, where $type(h)$ specifies the type (domain) of h. We refer to a single configuration of H as the actor's (hypothesized) 'mental state.' The allowable variable types (and the allowable number of variables of each type) are specified by the observer's Level 3 framework theory. For now, we shall consider the following variable types:

   - Discrete (DIS) variables take values in $\{1, \ldots, n\}$.
   - Continuous (CON) variables take values in $\mathbb{R}^n$, and are typically reserved for parameters.
   - Symbolic (SYM) variables serve as pointers to possible features in the environment. Each SYM variable X is associated with a primitive comparison function $X(s, v)$, where s is a system state and v is a possible value of X. This function returns *true* if state s includes the feature v that X "points to," and false otherwise. For example, we can define the goal "get a cookie" using a SYM variable which returns *true* in any state where the actor possesses a cookie, and false otherwise. If an actor model includes a SYM variable, the observer may use the associated primitive function in constructing the programs for that model. We use SYM-L to denote a list of SYM variables.
   - Representational (REP) variables take values in some space of representations of the world. These representations inherit their structure from the observer's MDP representation of the world itself. The value of a REP variable can be interpreted as a set of possible (or counterfactually possible) world states. For example, the observer could represent an actor's uncertainty about the current environment using a REP variable which captures the set of possible

world states compatible with what the actor has already observed so far. Usually, we can compactly encode this using an "underspecified" world state, by taking the same feature structure we use in the MDP representation of the world, and adding a marker for "missing" feature values (corresponding to those parts of the world the actor is not aware of).

- Valuative (VAL) variables denote value functions. The domain of a VAL variable may be any combination of observable states (e.g. system states, actions, sequences of actions, etc.). For example, a more complex rational action model may encode the actor's goals using a VAL variable whose values are utility functions over states of the world.[21]

In addition to types, each variable is assigned a temporal signature which specifies the scope of its persistence. Allowable temporal signatures are

- t, where x_t denotes that the value of x is specific to the current trial iteration,
- $0, T$, where x_0 denotes a feature specific to the initial state of a trial, and x_T denotes a feature specific to the final state of the trial,
- (no subscript), where x denotes an actor feature/parameter that is persistent across the trial

2. An *update function* $uf(system, state, H; \Theta_u)$, with parameters in Θ_u. This is a probabilistic program which accepts as inputs a system definition, a state value

[21]Technically we can encode SYM and REP variables as VAL variables, but it will be conceptually and notationally convenient to distinguish between the three

within that system, and a current mental state, and returns an updated mental state. If the H argument is empty, uf returns an *initial* mental state.

3. An *action function* $af(system, state, H; \Theta_a)$, with parameters in Θ_a. This is a probabilistic program which accepts as inputs a system definition, a state value within that system, and a current mental state, and returns an action within that system[22]

4. A parameter vector $\theta = \{\theta_u, \theta_a\}$, where $\theta_a \in \Theta_a$ and $\theta_u \in \Theta_u$.

The structure(s) and parameter space(s)[23] of the actor model are determined by the observer's Level 3 framework theory.

An actor model $\mathcal{M} = \{H, uf, af, \theta\}$ consists of these four components. Importantly, note that these components are not quite sufficient to specify a generative model, as they are defined without reference to trial-specific observations (i.e. they do not contain variables corresponding to actual observations). Rather, an actor model provides the "ingredients" necessary to construct a generative model of a particular trial, by associating each observed state and action variable with the generative functions defined above. To illustrate this, we provide and explain three simple examples of actor models

[22]We shall assume that the observer knows $states\_toActs$ and T for any system in which they have data

[23]The Level 3 theory may either specify a) a single model structure (and its corresponding parameter space) and a prior over that parameter space, or b) a set of possible model structures (and their corresponding parameter spaces), a prior over model structures, and a prior over each possible parameter space. We illustrate both kinds of inference in the next section

(programs are shown in a highly condensed, intuitive psuedocode), as well as a condensed graphical representation of each model structure.

Model specifications:

$M : \{(g, SYM)\}$

$uf(m, s)$:
    return (m ==undefined)? $categorical(vals$: $objects(s), probs$: $\beta)$: m

$af(m, s)$:
    return $flip(\epsilon)$? $randomMove(s)$: $shortestPath(m.g, s)$

$\Theta = \{\beta, \epsilon\}$

Graphical Representation:

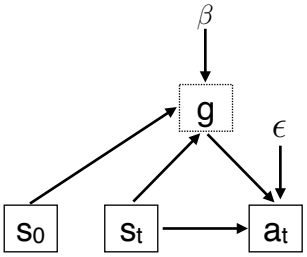


Figure 3.4.1: "error-prone fixed-goal" actor model

Figure 3.4.1 depicts the actor model used to construct the trial explanation in figure 3.3.5. This model includes a single mental state variable g, which points to a single possible feature in the environment. According to this actor model, the variable g is sampled at the start of the trial, and its value is persistent throughout the trial. The distribution from which g is sampled corresponds to a categorical distribution over accessible objects from the trial's initial state ($objects(s_0)$), with probabilities determined by the actor's preference parameter vector β (renormalized to outcomes in $objects(s_0)$). The helper function $shortestPath(m.g, s)$[24] computes the shortest path from the current system state s to any state in which the goal feature $m.g$ is satisfied, and returns the first step of that path (if the goal feature is satisfied in the current state, $shortestPath$ returns a terminal action which ends the trial). However, this actor model is "error prone:" the action function af returns an optimal action with probability $1 - \epsilon$,

[24]We can code the $shortestPath$ function in a system-general way by including T and $states\_toActs$ as inputs to the function

or a random action with probability ϵ.

Model specifications:

$M : \{(g, SYM)\}$

$uf(m, s)$:
    return $(flip(\nu) : m ==$undefined?$)$ $categorical(vals$: $objects(s), probs$: $\beta) : m$

$af(m, s)$:
    return $shortestPath(m.g, s)$

$\Theta = \{\beta, \nu\}$

Graphical Representation:

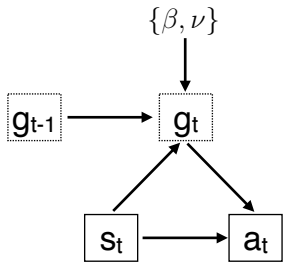


Figure 3.4.2: "Mind-change" actor model

Figure 3.4.2 depicts the actor model used to construct the trial explanation in figure 3.3.6. Like the previous model, this includes a single mental state variable pointing to a single feature in the environment. Unlike the previous model, the goal feature is not persistent throughout the episode, but may be resampled with probability ν each time the mental state is updated. This also differs from the previous model in that this actor is not error prone (i.e. will always output an optimal action for the current goal).

Model specifications:

$M : \{(g, SYM), (b, SYM\text{-}L)\}$

$uf(m, s)$:
    $bNew = addLineOfSight(s, m.b, range)$
    $gNew = (bNew == m.b)?\ m.g$: $categorical(vals$: $objects(bNew), probs$: $\beta)$
    return $\{bNew, gNew\}$

$af(m, s)$:
    return $shortestPath(m.g, s)$

$\Theta = \{\beta, range\}$

Graphical Representation:

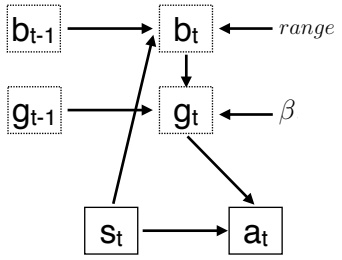


Figure 3.4.3: "awareness-constrained" actor model

The model in figure 3.4.3 includes a second mental state variable b, which tracks the

actor's "awareness" state and has type SYM-L (i.e. a list of symbolic pointers). Under this model, the update function first updates the actor's awareness state to the current system state. This involves a "line of sight" function which scans the actor's line of sight and constructs a pointer for each object that is encountered, which is then added to the list of pointers in $m.b$ (i.e. the full list of objects that the actor has previously encountered). If this list is changed in the process (i.e. if the actor encounters a new object), the actor resamples their goal state from among those features in the actor's current awareness state (i.e. chooses a goal object from among those known to be in the environment). The line of sight function includes a "range" parameter, which determines how far the actor's line of sight extends.

3.4.2 Inferring an actor model

A Level 2 problem involves inferring an actor model for a particular actor. The data consist of multiple trials involving that particular agent, and may span across multiple systems. Figure 3.4.4 shows an example of such a data set, including two trials from different grid world systems, and a third trial from a distinct "vending machine" system.

In addition to the data, the other input for Level 2 inference is the actor's framework theory, which we can characterize as a prior distribution $P(\mathcal{M})$ over possible actor models. The set and types of possibilities over which this prior distribution ranges is determined by the scope of the problem and depth of the observer's background knowledge, but the support of $P(\mathcal{M})$ will have one of the following types:

- A set of possible model structures (i.e. the mental variables and unparameterized forms for af and uf), a prior distribution over model structures, and a prior

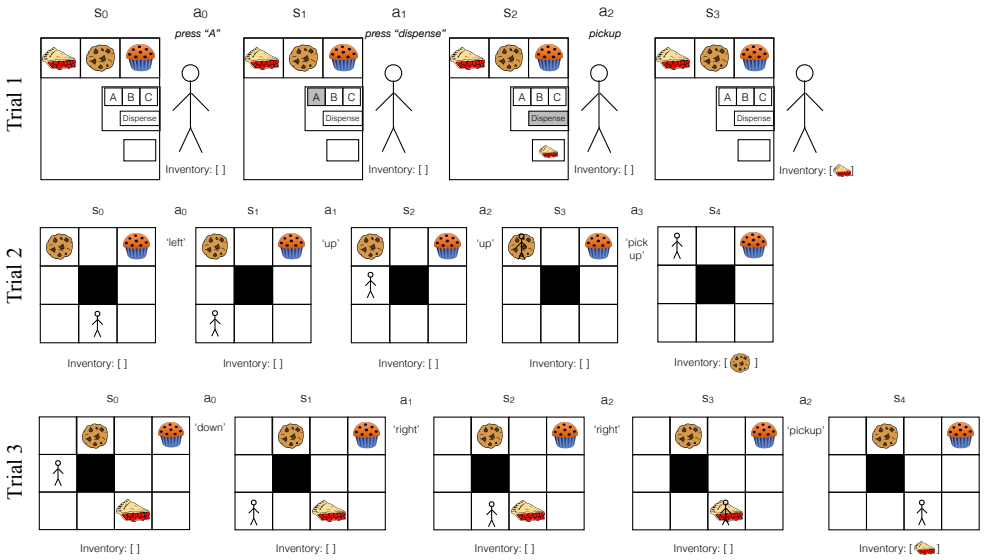


Figure 3.4.4: Example of cross-trial data for a single actor

distribution over the parameter space for each structure. Depending on the observer's background knowledge, this may be a rich set of possible structures, or a handful of specific structures, or a single feature of a structure that is otherwise fixed.

- A set of parameterizations for a single model structure (for some or all of the model's parameters)

For example, if the observer believes that all actors behave according to the same set of programs, but differ in terms of the parameter values for these programs, then the observer's Level 2 inference will involve inferring parameter values only. If the observer believes that actors are varied in their cognitive constraints (e.g. some actors engage in complex planning, others act impulsively on basic drives; some actors can see through walls, others cannot, etc.), then their Level 2 inference problem will involve both

identifying the structural constraints (i.e. mental variables and program forms) and the parameter values for that model. In the next section, we elaborate on how to construct and infer these prior distributions. For now, we shall consider two simple examples, one of each type.

For the first example, suppose our observer's framework theory includes a single model structure corresponding to the"error-prone fixed-goal" model in figure 3.4.1. This model has parameters $\theta = \{\beta, \epsilon\}$, where

- β is an $n \times 1$ parameter vector encoding the value that the actor would derive from obtaining each of n possible outcomes. In the uf function for this model, the vector β is restricted to include only those values corresponding to outcomes which are attainable in the current system. Let $outcomes(s)$ denote the set of possible outcomes attainable from system-state s, and $\beta_s = \{\beta_i \in \beta | i \in outcomes(s)\}$ (i.e. the parameter values associated with attainable outcomes). Intuitively, the values in β_s encode the likelihood that the actor will choose any one particular outcome from those attainable in the current system (though in general, the exact function that translates these values into action probabilities depends on the actor model).

- $\epsilon \in (0, .5)$ is an error parameter, which encodes the probability that the actor will make a mistake (i.e. choose a suboptimal action).

In this case, the observer's framework theory $P(\mathcal{M})$ (prior distribution over models) can be written as $P(\theta) = P(\beta, \epsilon)$, since $P(\mathcal{M})$ assigns all probability mass to a single model structure with parameters β and ϵ. Similarly, since the framework theory fully specifies the model structure, the problem of inferring an actor model only involves inferring the values of β and ϵ.

To illustrate how this inference works, let $D = \{d_1, \ldots, d_n\}$ be a set of trial observations for this particular actor. In order to infer the parameter values, the observer must compute the posterior distribution $P(\theta|D, M)$,[25] where M denotes the fixed model structure determined by the framework theory. On a computational level, this can be done via Bayes' rule:

$$P(\theta|D, M) \propto P(D|\theta, M)P(\theta) \tag{3.4}$$

The second term on the right hand side $P(\theta)$ is given by the observer's framework theory. The first term is the data likelihood. Because each trial is conditionally independent of each other trial (conditioned on the actor model), this term decomposes into

$$P(D|\theta, M) = \prod_{i=1}^{n} P(d_i|\theta, M)$$

and each $P(d_i|\theta, M)$ decomposes into a product of terms of the form $P(obs^i_j|\theta, M)$, where obs^i_j denotes the actor's jth action in the ith trial. In order to compute each $P(obs^i_j|\theta, M)$, the observer must marginalize out the hidden states in H (or approximate this marginalization with a single MaP estimate for each hidden state). In this case there is only one hidden variable g with values in $outcomes(s^i_j)$, so the computation reduces to

$$P(obs^i_j|\theta, M) = \sum_{v \in outcomes(s^i_j)} P(obs^i_j|\theta, M, g = v)P(g = v|\theta, M)$$

The action probability $P(obs^i_j|\theta, M, g = v)$ is given by the action function af, and the

[25] or a Maximum a Posteriori (MaP) estimate

prior probability $P(g = v|\theta, M)$ is given by the corresponding entry in parameter β.

On an algorithmic level, the observer can perform this computation using the following program form.[26]

```
var Level2_paramInference =function(data, theory){
    return Infer(function(){
        var θ = sample(theory.prior)
        var actorModel = extend(theory.structure, θ)
        var observeTrial =function(trial){
            var simTrial = actorModel(trial[0])
            Condition(simTrial == trial)
        }
        map(observeTrial, data)
        return θ
    })
}
```

The *theory* input contains the observer's framework theory, which in this case consists of a single model structure *theory.structure* and a prior distribution *theory.prior*. The model inside this Infer operator first generates a value of θ by sampling from the prior, then creates the corresponding actor model via $extend(theory.structure, \theta)$, which fills in the missing parameter values with the newly sampled θ. The function $observeTrial(t)$ generates a simulated trial using the same initial state as trial t, then adds the condition that this simulated trial is identical to the observed trial. The $map(f, v)$ operator applies a function f to each element of vector v, so $map(observeTrial, data)$ applies $observeTrial$ to each trial in *data*, thereby conditioning the distribution on observing all of the data. If we let $T = \{M, P(\theta)\}$ denote the observer's framework theory, then a call

[26]This pseudoprogram is theoretically sound and conceptually clear, but generally intractable. See appendix B for the actual program and details on how to make it tractable

to $Level2\_paramInference(D, T)$ returns a distribution object corresponding to the posterior distribution over actor models $P(\theta|D, T)$. Note that the hidden-state marginalization is performed implicitly by $actorModel()$, which returns a complete trial including simulated values for all hidden variables in the model.

Now, suppose that the observer's framework theory includes two model structures M_1 and M_2, corresponding to the actor models shown in figures 3.4.2 ("mind-change") and 3.4.3 ("awareness-constrained"). Note that these models induce different parameter spaces: $\theta_1 = (\beta, \nu)$, where $\nu \in (0, 1)$ is the actor's "fickleness" parameter, and $\theta_2 = (\beta, \rho)$, where $\rho > 0$ is the actor's visual range parameter. In this case, the observer's framework theory $P(\mathcal{M})$ can be written as

$$P(\mathcal{M}) = P(M, \theta) = P(M)P(\theta_M)$$

where $P(M)$ is the prior probability of a particular model structure, and $P(\theta_M)$ is a prior distribution over the parameter space associated with model structure M.

On a computational level, the observer's inference problem takes the same form as before (equation 3.4), but the inference and prior now range over (M, θ) instead of just θ:

$$P(M, \theta|D) \propto P(D|M, \theta)P(M, \theta) = \prod_{i=1}^{n} P(d_i|M, \theta)P(M)P(\theta_M) \tag{3.5}$$

The term on the right hand side requires the same series of computations as in the previous example (where the observer was only inferring the parameter value), but this computation is performed for each possible model structure and scaled by the structure prior $P(M)$.

On an algorithmic level, the observer can perform this computation by modifying the previous program form:

```
var Level2_structureInference =function(data, theory){
    return Infer(function(){
        var modelStructure = sample(theory.structurePrior)
        var θ = sample(theory.paramPriors[modelStructure])
        var actorModel = extend(modelStructure, θ)
        var observeTrial =function(trial){
            var simTrial = actorModel(trial[0])
            Condition(simTrial == trial)
        }
        map(observeTrial, data)
        return θ, modelStructure
    })
}
```

In this case, the *theory* input consists of a prior distribution over model structures *theory.structurePrior*, and a *function theory.paramPriors* which maps a model structure to a prior distribution over the corresponding parameter space. This program first samples a model structure, then samples a parameter vector for that model, then conditions the distribution on observing each data trial being generated by the sampled actor model. The resulting distribution object encodes the posterior distribution $P(M, \theta|D)$. Alternatively, we can replace the final line to only return the *modelStructure* variable, in which case the program marginalizes out the parameters.

Due to the inherent modularity of functional PPLs, we can define a single general-purpose program for performing any Level 2 inference problem (shown below). Here, the *theory* input consists of the structure prior *theory.structurePrior*[27] and a map from structures to parameter priors *theory.paramPriors*(*structure*). The *query*

[27]if the observer's theory specifies a single model structure, then we code

input specifies which feature of the actor model to return (e.g. full structure, single structural feature, full parameter set, single parameter, etc.). Thus, a call to the program $Level2\_Inference(data, theory, query)$ returns the posterior distribution $P(query|theory, data)$.

```
var Level2_Inference =function(data, theory, query){
    return Infer(function(){
        var modelStructure = sample(theory.structurePrior)
        var θ = sample(theory.paramPriors[modelStructure])
        var actorModel = extend(modelStructure, θ)
        var observeTrial =function(trial){
            var simTrial = actorModel(trial[0])
            Condition(simTrial == trial)
        }
        map(observeTrial, data)
        return getQuery(actorModel, query)
    })
}
```

3.5 Level 3: details

3.5.1 Framework Theories

The observer's framework theory captures general knowledge and expectations about agent behavior, abstracted away from trial-specific or actor-specific observation. Intuitively, Level 3 problems can include

- Inferring a general heuristic/structural template for generating behavior. For example, the "belief-desire-rational action" heuristic often attributed to human folk psychology can be formalized as a psychological framework theory.

theory.structurePrior as a Delta distribution concentrated on that structure

- Inferring general properties about actor's preferences/values. For example, determining whether there are certain outcomes which actors are generally more likely to prefer, or determining whether actors tend to have "peaked" preferences (i.e. highly concentrated around a small number of possible outcomes) or "flat" preferences (i.e. evenly distributed among a large number of outcomes).

- Identifying distinct actor "types" or "categories," and the general heuristics and properties of each type.

Formally, we define a framework theory T as a prior distribution $P(\mathcal{M})$ over actor models, which is used to do posterior inference in Level 2 as illustrated in the previous section. There are several forms that this prior could take, depending on the scope of the observer's inference problem and higher level knowledge/constraints.[28] We shall focus on three possible forms, which we present in order from most restrictive to least:

1. A single model structure M, and a prior distribution $P(\theta)$ over the associated parameter space. For example, the framework theory may specify a single belief-desire-action heuristic, parameterized by the actor's preferences, values, error rate, etc. In this case, the observer's framework theory specifies a prior distribution over these parameters.

2. A *mixture* of model structures $(M_1, \ldots, M_n)$, with an associated prior $P(M)$ over structures, *and* a prior distribution $P(\theta_i)$ over each corresponding parameter space. For example, the framework theory may specify that there are two types of agents: rational goal-seekers who may change their minds midway through a trial, and

[28] We discuss these constraints more explicitly in the next section

goal-seekers who never change their minds during a trial (but may make a mistake on their way to the goal). In this case, the framework theory would specify the prior probability that a new actor would have one of these two models, and a prior distribution over the parameter space for each model. We can also use a non-parametric prior such as a Dirichlet Process to define an *infinite mixture model* (Rasmussen 2000), which allows the observer to learn the number of categories directly from data.

3. A *probabilistic generative grammar* (PGG) for model structures (and associated parameter priors). PGGs use stochastic recursion to define compositional prior distributions over structures with arbitrary complexity, and are often used for Bayesian semantic parsing (e.g. Saparov & Mitchell 2016), and other Bayesian inference tasks over dynamic/sequential data (e.g. Nakamura et al 2016). A PGG framework theory is useful for cases where there is no obvious upper limit on the complexity of a model. For example, the observer may believe that actors are capable of forming hierarchically nested plans (i.e. a high level goal entails a sequence of sub-goals, each of which may itself require a sequence of sub-sub-goals, etc), but be uncertain about how many hierarchical levels an actor's plan may have. We can define a prior distribution over arbitrarily complex hierarchical goal structures using a PGG for "plans."[29]

While the PGG construction is clearly the most general and flexible, it is also the most computationally demanding. In this dissertation, we provide simulations and examples using the first two types of construction, though we outline a simple PGG for actor

[29]We provide a simple example of such a PGG in appendix B4

models in Appendix B4.

3.5.2 Inferring a framework theory

The data for Level 3 inference comprise multiple trial observations for multiple different actors (ideally across multiple different systems). We write $\mathbb{D} = \{D_1, \ldots, D_n\}$ to denote a data set containing trial observations for n different actors, where each $D_i = \{d_1^i, \ldots, d_{m_i}^i\}$ contains a trial observation for each of m_i trials involving actor i. An example of such a data set is shown in figure 3.5.1.

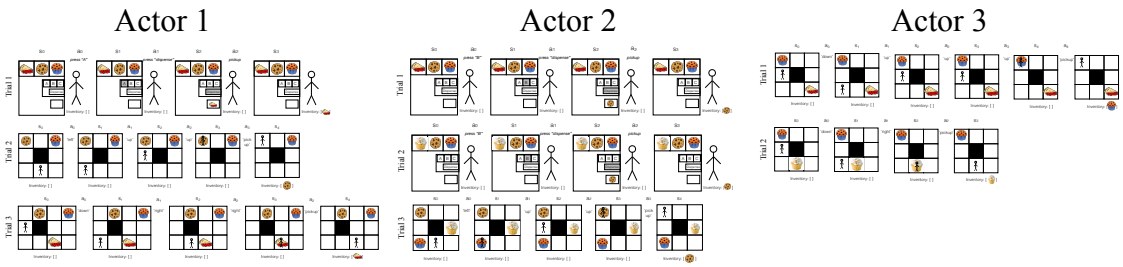


Figure 3.5.1: Example of cross-actor data for Level 3 inference

In addition to the data, Level 3 inference is constrained by a higher-level *over-hypothesis*. From a modeling perspective, the over-hypothesis defines the highest level of the observer's background knowledge which we take to be fixed throughout inference.[30] This knowledge includes the form of the framework theory $P(\mathcal{M})$ (for our purposes: "single-model," "mixture-model," or "PGG"), a space $\mathbb{M}$ of possible actor

[30]Of course, as with any hierarchical model, we can always construct a higher-level model for inferring this higher-level over-hypothesis, which would itself be constrained by an even higher-level over-hypothesis. So this is only the "highest level" of background knowledge in the sense that we take it as given in Level 3 inference

models, a space $\mathbb{P}$ of *prior distributions over* this space of models, and a prior distribution over $\mathbb{P}$. An over-hypothesis $\mathbb{T}$ with these components defines a prior distribution $P(T)$ over framework theories. Furthermore, since each T is itself a prior distribution over actor models, we can also construe $\mathbb{T}$ as a *hyper-prior* for actor models, i.e. a prior distribution *over* prior distributions over actor models.

On a computational level, a Level 3 problem involves computing the posterior distribution $P(T|\mathbb{D}, \mathbb{T})$, where $\mathbb{T}$ denotes the observer's over-hypothesis, and $\mathbb{D}$ denotes a cross-actor data set. Before explaining this in more detail, it is important to pause here and clarify a potential confusion: recall that in the previous section, we define Level 2 inference as computing the posterior distribution over actor models $P(\mathcal{M}|D, T)$, where D is an actor-specific data set and T is a framework theory. In that context, we defined the framework theory to be a *prior* distribution $P(\mathcal{M})$ over actor models. Here, however, we refer to the observer's framework theory as the *posterior* distribution $P(\mathcal{M}|\mathbb{D}, \mathbb{T})$. This apparent discrepancy is resolved by the context of inference: when doing Level 2 inference, the framework theory is assumed to be fixed, and is therefore treated as "prior knowledge." At Level 3, however, the framework theory is itself the subject of inference. Thus, we call the framework theory a "prior distribution" when referring to it as an *input* to Level 2 inference, and we call it a "posterior distribution" when referring to it as the *output* of Level 3 inference.

With this clarification, we provide a general computational-level definition of Level 3 inference, before illustrating its various forms in more detail. To this end, let $\mathbb{T}$ denote an over-hypothesis which entails a prior distribution $P(T|\mathbb{T})$ over framework theories. Let $\mathbb{D} = \{D_1, \ldots, D_n\}$ denote a cross-actor data set. The observer's Level 3 inference

problem is to compute the posterior distribution

$$P(T|\mathbb{D},\mathbb{T}) \propto P(\mathbb{D}|T)P(T|\mathbb{T}) \tag{3.6}$$

If we assume that each trial depicts only one actor (i.e. different actors do not interact with/encounter each other during trials), then each actor's behavior is conditionally independent of each other actor's behavior given the framework theory T. We can therefore decompose the likelihood term $P(\mathbb{D}|T)$ into a product of actor-specific terms:

$$P(\mathbb{D}|T)P(T|\mathbb{T}) = \prod_{i=1}^{n} P(D_i|T)P(T|\mathbb{T})$$

In order to compute the likelihood $P(D_i|T)$ for a single actor, the observer must marginalize out the actor model,[31] i.e.

$$P(D_i|T) = \sum_{\mathcal{M}_i \in supp(T)} P(D_i|\mathcal{M}_i)P(\mathcal{M}_i|T)$$

where $supp(T)$ denotes the support of the distribution $T = P(\mathcal{M})$. Importantly, note that the likelihood term $P(D_i|\mathcal{M}_i)$ is the same likelihood that is computed during Level 2 inference. This will have two useful implications for inference when we get to the algorithmic-level presentation: first, we can re-use the computational and algorithmic machinery of Level 2 to construct our Level 3 programs. Second, we can easily code our Level 3 programs to perform Level 2 inference over the observed actors

[31] in practice it is more efficient to approximate this marginalization with a MaP estimate of $\mathcal{M}$ for each actor

contemporaneously. That is, we can define a single general-purpose program that infers a general posterior distribution over framework theories, given data from actors $A_1, \ldots, A_n$, and *simultaneously* infers a posterior distribution over actor models for each actor in the data. Adding this extra level of inference will often speed up Level 2 learning, through an effect sometimes referred to as the "blessing of abstraction" (Goodman et al 2011).

With that being said, the general algorithmic form for Level 3 inference is as follows:[32]

```
var Level3_Inference =function(data, overHyp, query){
    return Infer(function(){
        var theory = sampleTheory(overHyp)
        var makeActorModel = mem(function(actorID){
            return sample(theory)
        })
        var actorModels = map(makeActorModel, data.actorIDs)
        map(map(observeTrial(actorModels[i], t), data.trials[i]), actorIDs)
        return getQuery(theory, query)
    })
}
```

In this program, *data* consists of a list of actor IDs ($data.actorIDs$), and a set of trial observations for each actor ($data.trials[i]$). The $overHyp$ input encodes the observer's over-hypothesis, either as a fixed model structure and hyper-prior over its parameter space, or a mixture of model structures and a prior distribution over structures (and their parameter spaces), or a generative grammar for models and corresponding hyper-priors. The $sampleTheory$ helper function executes the stochastic process encoded in $overHyp$ to generate a sample from the appropriate hyper-prior. The

[32]The usual disclaimer applies: this pseudoprogram is theoretically sound and conceptually clear, but hopelessly intractable outside of simple cases. See Appendix B for actual programs

makeActorModel function generates an actor model for a particular actor by sampling from the prior distribution encoded by *theory*.[33] This function is mapped over the set of actor IDs in *data* to create an actor model for each actor (stored in the array $actorModels$). The $observeTrial(actorModel, trial)$ function adds the Condition that the data in *trial* were generated by $actorModel$, and the nested $map(map(observeTrial \ldots$ statement applies $observeTrial$ to each trial for each actor model, thereby conditioning the distribution on *data*. Finally, *getQuery* extracts the queried information for this inference. Possible queries include

- The predictive posterior over actor models (i.e. a new actor model sampled for a previously unknown actor about whom nothing has been observed)
- The predictive posterior over one particular feature or parameter among actor models
- The posterior distribution over actor models for each actor in the data. Note that this has the same outputs as performing Level 2 inference for each actor individually.

To help make this very abstract inference more concrete, we provide a very simple example before moving onto the next section. For this example, suppose the data $\mathbb{D}$

[33]The *mem* command is a memo-ization operator, which allows the program to create persistently random features; i.e. when $mem(f(x))$ is called the first time with input v, it will execute the probabilistic program $f(v)$. When $mem(f(x))$ is called a second time with the same input, it will return the same value it returned previously on that input, without executing $f(v)$. See appendix A for a more in depth explanation of memo-ization

comprise trial observations from multiple actors in a single "vending machine" system. This system is very simple: in each trial, the actor is presented with a list of options, and the actor's single action in each trial corresponds to choosing one of the options. The observer's over-hypothesis $\mathbb{T}$ specifies a single model structure, defined by a single mental variable g and the following programs:

$uf(s, m)$:

    return m ==undefined? $categorical(vals$: $objects(s), probs$: $\beta)$: m

$af(s, m)$:

    return $pressButton(s, m.g)$

Intuitively: $uf(s, m)$ first outputs a goal object from among those available in the current machine, with probabilities given by a value parameter vector β (restricted to the current possibilities). $af(s, m)$ then outputs the button corresponding to that goal object.

Since this over-hypothesis includes only a single model structure with single parameter vector β, a framework theory T consistent with $\mathbb{T}$ corresponds to a prior distribution over preference vectors $P(\beta)$. Thus, $\mathbb{T} = P(T)$ must specify a hyper-prior over β (i.e. a prior over priors over β). For this example, we define this hyper-prior using a symmetric Dirichlet distribution $Dir([[1]])$ (where $[[1]]$ denotes an $n \times 1$ vector of 1s, and n is the number of possible outcomes) and a *concentration parameter* $\gamma \sim \Gamma$ drawn from a Gamma distribution. Intuitively, $Dir(\rho)$ defines a distribution over probability vectors of length n; when ρ is a vector of 1s, this distribution is uniform over all possible probability vectors. Scaling ρ up (> 1) concentrates the distribution on probability vectors that are flat and dense (i.e. distribute probability mass close to evenly among lots of outcomes), while scaling ρ down (< 1) concentrates the distribution on probability

vectors that are peaked and sparse (i.e. distribute most of the probability mass among a small number of outcomes). Thus, if we separate the parameter ρ into an $n \times 1$ vector of 1s and the scalar concentration parameter γ, we can define a prior over preference vectors with a trainable concentration parameter, i.e. $P(\beta; \gamma) = Dir(\beta; \gamma * Dir([[1]]))$. In conjunction with the Gamma prior over γ, this defines a prior distribution over *prior distributions* over β, and each particular value of γ defines a prior distribution over β, which, in this case, corresponds to a particular framework theory.

Intuitively, this over-hypothesis reflects that the observer has no expectations about the distribution of a new actor's preferences (i.e. no prior expectations about which outcomes that actor is likely to prefer), but the observer does have prior expectations about how specific (peaked) or general (flat) the new actor's preferences will be. The Level 3 inference problem for this model is to compute the posterior distribution $P(\gamma|\mathbb{D}, \mathbb{T})$, which measures the likelihood that a new actor will have peaked or flat preferences, generalized from observations of other actors.

At an algorithmic level, the observer can perform this inference using the following program form

```
var inferGamma =function(data, hyperPrior, modelStructure){
    return Infer(function(){
        var gamma = sample(hyperPrior);
        var makeActor = mem(function(actorID){
            var beta = sample(Dirichlet(gamma * ones(N)))
            return extend(modelStructure, beta)
        })
        var actorModels = map(makeActor, data.actorIDs)
        map(map(observeTrial(actorModels[i], t), data.trials[i]), data.actorIDs))
        return gamma
    })
}
```

This program takes the same form as the previous, general Level 3 inference program: it first samples a value of the concentration hyper-parameter γ. The function *makeActor* samples a random actor model from the prior $P(\mathcal{M})$ entailed by hyper-parameter γ. In this case, it generates a random preference parameter vector and fills in the missing parameter from *modelStructure* with this value. The $actorModels$ object is constructed by sampling an actor model for each actor in the data. The $observeTrial(actorModel, trial)$ helper function adds the condition that the data in *trial* were observed in a trial generated by $actorModel$. The nested $map(map(\ldots$ command applies this function to each trial observed for each actor, thereby conditioning the distribution on the data. Thus, if we define the over-hypothesis $\mathbb{T} = \{hyperPrior, modelStructure\}$, then a call to $inferGamma(\mathbb{D}, \mathbb{T})$ returns the posterior distribution $P(\gamma|\mathbb{B}, \mathbb{T})$. This posterior distribution encodes the observer's updated expectations regarding how specific actors' preferences tend to be.

To illustrate this, we construct two sample data sets, each consisting of 10 vending-machine trials for each of 10 different actors. In the first data set ($\mathbb{D}_1$), actor

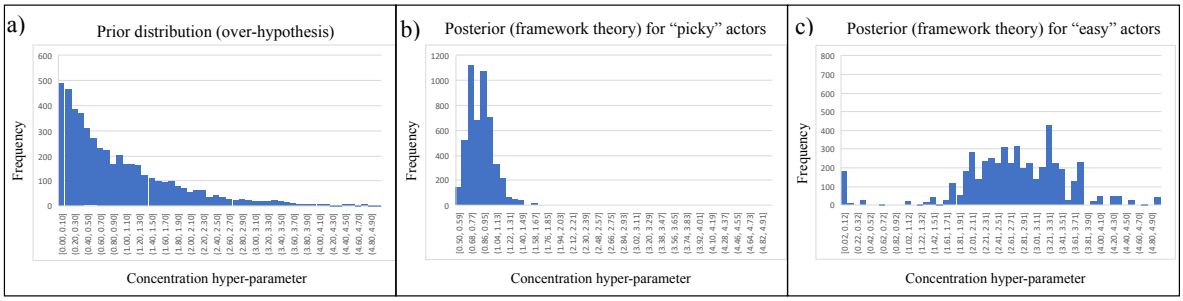


Figure 3.5.2: Results of Level 3 inference, shown as histograms. Panel a) depicts the prior distribution over γ (i.e. the observer's over-hypothesis). Panel b) depicts the posterior distribution (i.e. framework theory) inferred from a population of "picky" actors (i.e. dataset $\mathbb{D}_1$). Relative to the prior, this distribution is skewed towards low values, and reflects the knowledge that actors tend to be picky (i.e. have specific preferences). Panel c) depicts the posterior distribution inferred from a population of "easy" actors (i.e. dataset $\mathbb{D}_2$). Relative to the prior, this distribution is skewed towards high values, and reflects the knowledge that actor's tend to be easy (i.e. have general preferences)

models are generated by sampling a concentration parameter $\gamma \sim Uniform(.25, 1)$ (skewed towards low concentration values/peaked preference vectors), while the second data set ($\mathbb{D}_2$) is generated by drawing $\gamma \sim Uniform(.75, 5)$ (skewed towards high concentration values/flat preference vectors). We then apply the $inferGamma$ function defined above to each data set. The results are shown as histograms in Figure 3.5.2, which depicts the posterior distributions $P(\gamma|\mathbb{D}_1)$ and $P(\gamma|\mathbb{D}_2)$ compared to the hyper prior $\Gamma(1, 1)$. As expected, $P(\gamma|\mathbb{D}_1)$ is skewed towards large values of γ (relative to the prior) and $P(\gamma|\mathbb{D}_2)$ is skewed towards small values. Intuitively, this means that the observer has learned that the first population of agents tends to have general (flat) preferences, while the second population of agents tends to have specific (peaked) preferences.

3.5.3 What makes a framework theory "psychological"?

The definitions we provide in this chapter characterize framework theories in a very general way. Since lower level hypotheses (actor models and trial explanations) are ultimately defined in terms of the higher level theory, there is a wide range of possible representations and models compatible with these definitions. However, the ultimate purpose of this framework is to understand the development and application of a folk psychology, so what distinguishes the models we define here from a more general framework for causal learning and reasoning? To put this another way, under what conditions can we justifiably interpret a framework theory as "psychological," and how can we formalize these conditions as constraints on the very general space of theories we describe above?

There are two obvious constraints that stand out immediately. The first is pragmatic: a psychological framework theory must be useful for explaining, predicting, interpreting, or otherwise reasoning about human behavior. The second is more conceptual: the explanations provided by a framework theory must appeal to some kind of hidden state or value. However, these two constraints are still too general. To illustrate this point, it will help to consider a few examples of framework theories that are compatible with these constraints, but do not, we shall argue, seem "psychological" on an intuitive level. Two such theories are shown in figure 3.5.3. According to the "hive-mind" theory in figure 3.5.3a, a single pair of hidden states uniquely determines the behavior of multiple individual actors. This theory satisfies the two conditions we propose above: models in this theory clearly depend on posited hidden states, and, if appropriately parameterized, a model of this form could be used to provide something like a belief+desire+rational

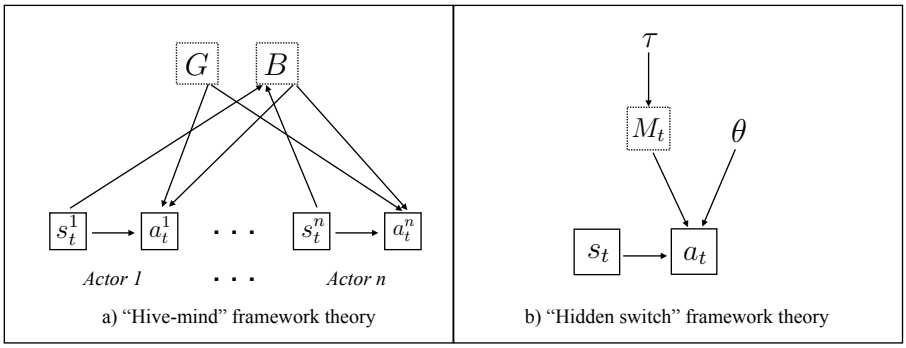


Figure 3.5.3: Two examples of framework theories that aren't necessarily "psychological"

action explanation of actor behavior, but for all of the actors collectively. But, intuitively, explanations of this form do not seem "psychological," as they explain each actor's behavior purely in terms of factors external to the actor. This points to another possible constraint on framework theories: the theory must attribute distinct and non-interacting hidden entities to individual agents.[34]

However, figure 3.5.3b shows a theory that satisfies this additional constraint as well, but does not seem to qualify as "psychological." According to this "hidden-switch" model, each actor is explained by a hidden binary state which is always set to "on" or "off." When switched to "off," the actor does nothing. When switched to "on," the actor behaves according to a categorical distribution over possible actions, parameterized by θ. This model fulfills all three criteria: given appropriate parameter values (i.e. an accurate estimate of the probability that the actor will take a certain action in a certain

[34]Alternatively, this sort of hive-mind/individualist distinction could be learned from data collected across different populations of agents (Though it is beyond the scope of the present dissertation). This inference would correspond to a "Level 4" (over-hypothesis) problem

system-state), a model of this form could provide predictive power over an actor's behavior; the model clearly relies on a hidden state, and the hidden state is specific to the actor. However, this model could also be straightforwardly applied to a simple electric toy with a hidden switch: when "on," the toy moves according to a fixed distribution; when off, the toy does nothing. Thus, even though models of this form are able to explain an actor's behavior in terms of actor-specific hidden states, the representational structure and causal role of this states make it intuitively difficult to recognize these explanations as "psychological."[35]

As these examples show, we must think carefully about what kinds of constraints on framework theories would allow us to justifiably interpret a theory as "psychological." Often, the constraints of our folk psychology are described and framed in terms of certain categories of mental states- typically values/goals/desires/etc. and beliefs/awareness/perspective/etc.- and certain kinds of functional relations between these states- typically some kind of utility calculus (e.g. Jara-Ettinger et al 2016) or principle of rational action (e.g. Csibra & Gergely 1995). Indeed, we spent much of chapter 2 arguing that, based on the available data, our folk psychology *does* appear to consistently involve this sort of "belief+desire+(rational) action" (BDA) heuristic. It is therefore tempting to derive our constraints from this apparent structure. That is, we could define "psychological framework theories" to consist of those theories (per our

[35]Alternatively, we may grant that any theory which attributes persistent, agent-specific hidden states is psychological, but distinguish between "intentional" and "non-intentional" psychological theories. In this case, the theory shown in 3.5.3b would be an example of a non-intentional psychological framework theory, in the sense that its only psychological state does not point to or represent anything outside the actor.

definitions in the previous two sections) which share certain structural, relational, and representational features with the folk psychology that does, in fact, consistently develop. We could interpret this as a set of structural constraints on the space of allowable theories, e.g. we could characterize what counts as a "value/goal/desire" variable and require that a psychological framework theory include such a variable.

Given our questions of interest, however, this would defeat the purpose of our analysis. After all, the high level questions motivating this dissertation are *why* and *how* do we so consistently develop a BDA folk psychology? It is clear that building this BDA structure into our high-level constraints would prevent us from answering these questions in any meaningful way. Instead, our approach appeals to the structure of the observer's inference problem to derive constraints. That is, we define constraints on framework theories in terms of

1. the data that the theory must explain (e.g. what kinds of environments/behavior/actors does the observer frequently encounter?),

2. the tasks that the theory must be used to solve (e.g. what does the observer need to explain? what aspects of behavior do they need to predict, and how accurately?), and

3. the observer's cognitive constraints (e.g. what are the observer's representational/computational/memory resources?)[36]

Our goal in chapter 5 is to show that BDA-like theories (or theories with BDA-like features) are, in some sense, the *rational* solutions to the mental inference problems we

[36]This is the where the PPL aspect of the framework becomes most useful

face in our everyday social experience. We therefore draw on constraints imposed by the task demands of mental and social inference, rather than the apparent structure of our (real-life) solutions to these problems. For example: we shall *not* observe that human psychological explanations are, in fact, framed in terms of hidden "goal" states (which have certain representational and relational properties), and then require that our framework theories include hidden states with these properties. Instead, we shall observe the structure of the social and mental inference problems we face in our everyday experience (including data, tasks, and the observer's cognitive constraints), and argue that the "rational" solutions to those problems are those that include hidden states with "goal-like" representational and relational properties.

To this end, in chapter 5, we shall use our framework to

1. formally define some of the mental and social inference problems we face in daily social experience,

2. formally characterize a notion of "rationality" that we can use to evaluate solutions to these problems,

3. show why (and in what circumstances) the "rational" solutions to these problems have a BDA-like structure, and

4. motivate an account of how these solutions could develop through domain-general inference processes (in conjunction with domain-specific prior knowledge)

In this sense, we shall argue that a) BDA heuristics provide a "rational" solution to social inference problems, and b) it is plausible that we develop BDA folk psychologies *because* they are rational solutions to the social inference problems we face. Before we

present these theoretical arguments, however, we first provide a methodological framework for connecting these computational models to cognitive behavioral data.

4 Connecting models with data

4.1 The challenges of infant cognitive studies

[37]In chapter 2, we described some of the main theoretical, conceptual, and methodological challenges involved in explaining cognitive development, and in chapter 3, we presented a computational modeling framework for addressing some of these challenges in the context of ToM development. In order for this framework to be effective, we will need to use it in two different ways: first, we will need to use it as a theoretical basis for demonstrating and justifying, through simulation, how (and under what circumstances) certain kinds of cognitive development could occur. Second, we will need to use it as a basis for interpreting behavioral data, and determining what that behavioral data reveals about our cognitive development. In this chapter, we present a methodological framework for the latter: that is, we demonstrate how our framework can be used to interpret data generated by cognitive behavioral studies. We focus in particular on infant cognitive studies for two main reasons: first, understanding the first 12-18 months of life is especially crucial for obtaining a complete picture of human cognitive development. Second, infant cognitive studies pose unique theoretical and methodological challenges, and we believe our framework is well suited to address these

[37]Portions of this chapter appear in the article *How do we know what babies know?* in the Journal of Philosophical Psychology (in press)

challenges.

One of the main reasons that infant cognitive studies are so challenging is that there are relatively few ways to collect relevant data. For obvious reasons, we cannot expect infants to tell us what they are thinking. For practical and technical reasons, standard neuroimaging techniques are difficult to apply to infants (Raschle et al 2012). Thus, we are often forced to rely on behavioral data alone. This presents a challenge in itself, as there aren't many behaviors that young infants can reliably perform. For this reason, the vast majority of infant cognitive studies use visual fixation time (i.e. the length of time that an infant visually attends to a stimulus) as the main measure of interest. This leverages one of the few behaviors that infants of all ages regularly engage in: staring at things.

There are many different methods for collecting fixation data from infants. Interpreting such data requires an assumption that connects fixation times to some underlying cognitive process; such assumptions are called *linking hypotheses* (Aslin 2007, Teller 1984). Intuitively, a linking hypothesis tells us what the infant's fixation behavior reveals about how the infant perceives or processes a stimulus. One of the most common looking time paradigms- the *visual habituation paradigm*- relies on a linking hypothesis that connects an infant's fixation time with the novelty, complexity, or unexpectedness of the stimulus. The origins of this assumption are often attributed to a series of studies performed by Robert Fantz in the late 1950s and early 1960s (Fantz 1958, 1961, 1964). In these studies, infants were shown a sequence of paired stimuli shown side-by-side. In each step, the stimulus on side one was held fixed, while the other stimulus varied between steps. The authors found that, across trials, infants gradually shifted their attention away from the fixed stimulus and towards the novel stimuli. The term

habituation refers to this gradual decrease in fixation time as a stimulus is repeated (and thereby becomes more familiar).

Since these findings, habituation experiments have been used extensively to investigate how infants represent and understand the world. This includes infants' understanding of object physics (e.g. Baillargeon 1986, Leslie 1984, Spelke et al 1994, Spelke et al 1995) how infants perceive and interpret causal events (e.g. Cohen & Oakes 1993, Leslie & Keeble 1987, Muentener & Carey 2010, Oakes & Cohen 1990), how infants interpret intentional actions (e.g. Brandone & Wellman 2009, Csibra & Gergely 1998, Gergely et al 1995, Paulus et al 2011, Phillips & Wellman 2005, Woodward 1998), and whether infants are sensitive to the beliefs of other agents (e.g. Brooks & Meltzoff 2002, Kovacs et al 2010, Onishi & Baillargeon 2005, Surian et al 2007). Indeed, without the development of looking time experiments, we would know very little about infant cognition whatsoever.

There are, however, a number of concerns regarding what these experiments reveal and how their results ought to be interpreted. First, the conclusions drawn in any looking time experiment depend critically on the assumed linking hypothesis, and it is therefore crucial to thoroughly test and understand the linking hypothesis itself. To this end, several authors have called for an increased focus on studying and formally modeling the habituation process per se (e.g. Aslin & Fiser 2005, Colombo & Mitchell 2009). Other authors have expressed more theoretical concerns about how hypotheses ought to be formulated and validated (Aslin 2007, Oakes 2010), and more practical concerns about the proper criteria for establishing when an infant has habituated to a stimulus (Dannemiller 1984, Thomas & Gilmore 2004).

More generally, any study that asks what a subject knows, or how a subject

represents a stimulus, faces a certain kind of underdetermination that can be difficult (or impossible) to resolve. In particular, there are often multiple distinct accounts of how a subject represents a stimulus that result in identical behavior under the same experimental assumptions. To illustrate this, consider the following scenario: two young siblings (Ivan and Amos) go with their parents to an ice cream shop and each get a cone. After leaving the shop, Ivan drops his cone and begins to cry. Now consider the following two descriptions of what transpires next:

1. Amos sees that his brother is crying because he dropped his cone. He doesn't like it when his brother cries, so he gives Ivan his own cone, hoping this will help Ivan stop crying.

2. Amos sees that his brother feels sad because he dropped his cone. He doesn't like it when his brother is sad, so he gives Ivan his own cone, hoping this will help Ivan feel better.

Both cases describe (from an outside observer's perspective) the same stimulus and response: Ivan drops his cone and begins to cry, and Amos gives Ivan his own cone. But the reasoning underlying Amos' response in each scenario is very different. The first explanation is strictly behavioral: Amos reasons that Ivan's behavior (crying) is a direct reaction to a change in his environment (dropping his cone). The second explanation is more mentalistic: Amos reasons that Ivan's behavior is a reaction to a hidden mental state (sadness) which is caused by dropping his cone. Both of these accounts describe the same sequence of events, but involve very different reasoning and representations. If Amos is sufficiently verbal, we might try to distinguish these two accounts by asking him to explain his reasoning, but that is not an option when the subject is a preverbal infant.

For this reason, great care must be taken to precisely specify the hypotheses being tested in a habituation experiment, and the behavioral predictions we generate from each hypothesis.

In this chapter, we will demonstrate how our computational modeling framework can be used to

1. precisely specify a set of hypotheses about how an infant represents a stimulus,
2. precisely specify the linking assumptions that relate cognitive representations to experimental behavior,
3. apply these linking assumptions to generate behavioral predictions from each hypothesis via a simulated version of habituation, and
4. evaluate which hypotheses can be validated or refuted by a given experiment.

In section 4.2, we start by reviewing visual habituation experiments in more detail, and identify the theoretical and methodological challenges which our framework addresses. We focus in particular on two main issues: the lack of formalizations of hypotheses about infants' cognitive representations, and the lack of formalizations of the linking assumptions through which experimental results are interpreted. In section 4.3, we illustrate how our framework provides the three components described above. In section 4.4, we illustrate these applications by replicating a seminal study on infants' understanding of intentional actions (Woodward 1998). We demonstrate how our framework allows us to formalize the qualitative question posed in this study, how it enables a more precise interpretation of its results, and what further insights or analysis this interpretation suggests.

4.2 Background

4.2.1 Looking times and habituation experiments

While infant fixation studies date back to the early 20th century (Segers 1936, Valentine 1913), the seminal demonstration of infant habituation is often attributed to a series of studies by Robert Fantz in the early 1960s (Fantz 1958, 1961, 1964). In the 1964 study, each infant was shown a sequence of stimuli paired side-by-side. In each trial, the stimulus on one side was held fixed, while the stimulus on the other side was novel. As the trials progressed, infants gradually shifted their visual attention away from the fixed stimulus and towards the novel one. *Habituation* refers to this progressive decrease in fixation time as a stimulus becomes more familiar.

Since this study, the logic of visual habituation has been developed into a set of experimental paradigms, which we briefly describe here. A standard habituation experiment[38] involves an initial habituation phase and a subsequent test or *dishabituation* phase. In the habituation phase, the infant is repeatedly shown the same stimulus over multiple trials, and the experimenter records the infant's *fixation time*; that is, the duration for which the infant attends to the stimulus before looking away. As the stimulus is repeated, the infant's fixation time progressively decreases. The habituation phase is typically ended once fixation time reaches some pre-defined threshold (Horowitz et al 1972), at which point the infant is assumed to be habituated and the test phase begins.

In the test phase, the infant is shown two or more stimuli, sometimes in sequence, sometimes simultaneously. Typically, the stimuli are designed so that the habituation

[38]Henceforth we use "habituation" to denote "visual habituation" specifically

stimulus contains one or more salient features, and the test stimuli each differ along one of these features. Dishabituation refers to a sharp increase in fixation time as the infant first encounters a stimulus that differs from the habituation stimulus. If the infant is shown two test stimuli (say, A and B), and dishabituates significantly more to one stimulus over the other (say, stimulus A), this is interpreted to mean that stimulus A appears more novel, unexpected, or complex to the infant, relative to the expectations formed during habituation. Under this reasoning, we can design the experimental stimuli to determine which features are most integral in the infant's representation of a stimulus.

To better illustrate this paradigm, recall the seminal habituation study into infants' understanding of goal-directed actions that we briefly described in chapter 2 (Woodward 1998). The habituation stimulus in this study (see figure 2.1.1) consists of a stage with two platforms, each holding a visually distinct toy. An actor stands to one side of the stage, so that only their arm is visible. In each habituation trial, the actor reaches for and grasps one of the toys, targeting the same toy each time. In the test phase, the position of the two toys is switched, while the actor remains on the same side. The infant is then shown two test events. In the "new-goal" test event, the actor performs the same physical reaching motion as in the habituation stimulus (long-reach or short-reach), thereby grasping the opposite toy. In the "new-motion" test event, the actor performs the opposite reaching motion, thereby grasping the same toy.

The logic underlying this design is that the habituation event depicts two salient features: *action* (spatiotemporal profile of the actor's arm), and *outcome* (which toy is grasped). If the infant encodes the habituation stimulus in terms of the action feature, then the test event which varies the action (new-action) should appear more novel to the infant, resulting in a higher rate of dishabituation. Conversely, if the infant encodes the

habituation stimulus in terms of the outcome, the new-goal test event should appear more novel. In this study (and a follow-up series of replications and control studies), 8- and 9-month old infants consistently dishabituated to the new-goal event at higher rates, leading to the conclusion that infants represent reaching actions in terms of the target object, rather than physical arm motion.

The Woodward (1998) experiments nicely illustrate the core principles underlying habituation studies. Similar methods have been used extensively to explore how infants represent different aspects of the world, such as object physics (e.g. Baillargeon 1986, Leslie 1984, Spelke et al 1994, Spelke et al 1995), causation (e.g. Cohen & Oakes 1993, Leslie & Keeble 1987, Muentener & Carey 2010, Oakes & Cohen 1990), intentions (e.g. Brandone & Wellman 2009, Csibra & Gergely 1998, Gergely et al 1995, Paulus et al 2011, Phillips & Wellman 2005, Woodward 1998), and beliefs (e.g. Brooks & Meltzoff 2002, Kovacs et al 2010, Onishi & Baillargeon 2005, Surian et al 2007). However, a number of authors both within and adjacent to developmental psychology have expressed concerns about the design and interpretation of habituation experiments, calling for an increased focus on studying the process of habituation itself, and an increased focus on computational models of habituation (Aslin 2007, Aslin & Fiser 2005, Colombo & Mitchell 2009, Oakes 2010, Thomas & Gilmore 2004).

4.2.2 Theoretical models of habituation

Infant fixation data can only be interpreted through the lens of a *linking hypothesis*, which specifies the underlying cognitive or neurological processes that drive infants to selectively allocate their visual attention. There have been multiple proposed accounts of what drives this process, but most accounts share a common view that infant looking

times reflect some combination of stimulus-driven attention, memory of previously encountered stimuli, and some means of comparison between past and present stimuli (Kidd et al 2012).

One of the longest-standing and most widely cited explanations for infant habituation is the Sokolov comparator model (Sokolov 1963). This model was based on observations of an orienting reflex (OR), a set of behavioral responses to nonthreatening, novel stimuli of moderate intensity (Colombo & Mitchell 2009). Research on the OR in animals demonstrated that the magnitude of the OR response would progressively decrease as a stimulus was repeated. Sokolov theorized that as an organism repeatedly encounters a stimulus, the organism forms an internal representation or "cognitive schema" of that stimulus, and compares the observed stimulus to the inferred representation. Under this theory, the magnitude of the OR response (in the context of infant studies, the infant's fixation time) is inversely proportional to the degree of similarity between observed stimulus and internal representation.

A second theoretical approach which became prominent in the 1980s is the dual-process account (Groves & Thompson 1970, Thompson & Spencer 1966). Under this account, the profile of an infant's fixation time in response to a repeated stimulus is determined by two independent processes. The first is a familiarization process, which is similar to the comparator model of habituation, and results in a similarly decreasing response to repeated stimuli. The second is a sensitization process, which induces a transient increase in response strength at the introduction of a new stimulus. Under a dual-process model, an infant's fixation time reflects both of these processes working in tandem. Several key predictions of this theory were confirmed in a series of infant studies in 1985 (Bashinski et al 1985).

Due to its explanatory power, relative conceptual simplicity, and plausible physiological underpinnings (Bernstein 1979, 1981), the comparator theory (or comparator + sensitization) has remained the dominant account of infant habituation, though many implications of dual-process accounts continue to be accounted for in experimental analysis. However, without formal models of certain key aspects, these theories provide only rough conceptual guidelines for designing habituation experiments, leaving unanswered many critical questions about proper experimental design and interpretation of experimental results.

4.2.3 Challenges and computational models

Here we review some of the concerns that have been raised about infant habituation studies. These include practical concerns about proper experimental design and physiological concerns about the neural substrates underlying habituation, and several authors have proposed computational models to help address these issues. However, our questions of interest relate more specifically to the "cognitive schema" or internal representations that an infant acquires during habituation, and what we can infer about these representations from the infant's fixation behavior. Several authors have identified key challenges in making such inferences about infants' cognitive representations (e.g. Aslin & Fiser 2007, Oakes 2010), and we argue that the current literature lacks the computational models necessary to address these challenges.

The first concern is that infants almost certainly form expectations and preferences for stimuli through their everyday experience, which may affect an infant's performance in an experimental setting. This is precisely the purpose of the habituation phase, during which the infant is repeatedly shown a "biasing stimulus" so as to shift the

infant's intrinsic expectations (either by eliminating a prior expectation, or inducing an expectation where none existed). However, infants' intrinsic expectations can still "seep through" to the post-habituation phase, making it difficult to determine whether the infant's test-phase fixation times solely reflect the expectations formed during habituation (e.g. Quinn et al 2002). Thus, great care must be taken to determine what pre-existing expectations might influence the infant's performance, and assess those expectations in order to accurately interpret experimental results.

Another concern is what we can conclude about the infant's internal representations when the habituation and test stimuli differ along inferred features as well as observable features. For example, if the habituation stimulus depicts a certain shape of a certain color, and the two test stimuli depict a) the same shape in a different color and b) a different shape in the same color, it is fairly straightforward to infer which feature (if any) is more integral to the infant's representation, as both features are easily perceptible. In many experiments, however, the novel and familiar stimuli differ in terms of some inferred feature, which has an observable effect but is not directly observable itself. A clear example is the Woodward (1998) experiment, wherein one of the test stimuli differs from the habituation stimulus in terms of the actor's goal. That infants consistently dishabituated more strongly to the "new-goal" test event clearly shows that infants can detect the physical features that are relevant to the actor's goal (i.e. the path and location of the actor's hand relative to the target object), but this does not directly show whether the infant represents those physical differences in the same way that an adult would (as observable consequences of the actor's hidden goal state). Thus, great care must be taken to precisely specify the hypotheses being tested, and to avoid drawing stronger conclusions (i.e projecting on the infant a richer mental representation)

than are warranted by the data.

As we cannot directly observe an infant's knowledge or internal representations, and we cannot ask infants to tell us about their internal representations, addressing these challenges requires a computational model that allows us to a) formulate precise hypotheses about how the infant represents a stimulus (and the relevant background knowledge that constrains this representation), and b) connect that hypothesis to a prediction about how the infant would act on a given representation. That is, because we cannot "observe" how the infant represents a stimulus, we must rely on principled counterfactual claims regarding how the infant *would* behave if they *did* represent a stimulus in a certain way, which we can then compare against their observed behavior in an experimental setting.

The current literature on formal models of habituation is largely characterized by two approaches, neither of which is very well suited for these tasks. One approach is to model habituation using regression analysis (e.g. Ashmead & Davis 1996, Dannemiller 1984, Thomas & Gilmore 2004), which is often used to perform robustness checks on certain experimental practices (e.g. the proper criteria for determining when an infant has habituated to a stimulus). These models abstract away from any details regarding infants' internal representations, instead treating fixation time as a function of trial time directly. The other approach uses connectionist and dynamic systems models to investigate the neurological substrates underlying habituation (e.g. Elman et al 1998, Sirois & Mareschal 2002, 2004, Van Overwalle 2010). These models deal directly with the low-level mechanisms involved in habituation, and are similarly unsuited for answering questions about what infants "know." Thus, there is a clear gap in the relevant literature at the level of cognitive representation, where our questions of interest

reside. In the next section, we argue that the framework we presented in chapter 3 is well suited for addressing questions about infants' knowledge and internal representations, and demonstrate how to connect those models with habituation data.

4.3 Methodological framework

In order to address the challenges described in 4.2, we need the three following components:

1. a way to formalize claims about how an infant represents a stimulus (i.e. the "cognitive schema" that an infant acquires),

2. a way of connecting these claims to predictions about infant looking behavior (i.e. a linking hypothesis), and

3. a way to model the process through which these representations are acquired during habituation.

We shall illustrate how the computational framework from chapter 3 provides these components.

4.3.1 Conceptual overview

At a high level, the purpose of this methodological framework is to enable us to make inferences about an infant's framework theory, based on their behavior in a two-stage habituation+test task. In order to make these inferences, we will rely on statements of the form "an infant with framework theory T, if habituated to stimulus s, will display fixation behavior v in response to test stimuli t_1 and t_2." We can use our computational

modeling framework to derive and justify such statements, and then invert these statements to make inferences about the infant's framework theory, based on their visual fixation responses in a habituation experiment.

In order to derive these statements, we need a way to characterize a set of possible framework theories that an infant may hold for a particular class of stimulus. To this end, we can use the Level 3 representations defined in chapter 3.5.[39] Each framework theory specifies a set of possible actor models that the infant may infer during habituation, and each actor model determines the expectations that the infant should display during the test phase. Given a class of experimental stimuli, we can constrain the set of plausible framework theories by appealing to what the infant is likely to already know. For example, there is evidence that young infants understand that causal relations respect the forward direction of time (Leslie & Keeble 1987), so we may omit any framework theory in which past features may be causally dependent on future features. We may also derive constraints from other levels of analysis (e.g. behavioral, physiological, etc.), though for this chapter we focus on constraints derived from expectations on the subject's background knowledge. The set of framework theories that are consistent with these constraints determines the set of hypotheses that we (i.e. the experimenters) may consider as plausible candidates for the infant's cognitive representations.

Given this set of plausible candidate theories, the next step is to characterize how the infant's framework theory determines the internal representation or "cognitive schema"

[39]Note that, for this application, we are not concerned with how the infant infers the framework theory to begin with, but how a given framework theory constrains the infant's behavior in a particular habituation task

they acquire during habituation to a particular stimulus. In the context of our computational framework, this corresponds to a Level 2 inference problem, i.e. inferring an actor model for the actor depicted in the habituation stimulus, from a single (repeated) instance of that actor's behavior. Thus, our Level 2 framework allows us to connect claims about the infant's framework theory with predictions about the cognitive schema the infant acquires during habituation. However, we obviously cannot directly observe the infant's cognitive schema, so the last step is to formalize a *linking hypothesis* that connects the observable data (i.e. the infant's fixation behavior during the test phase) to the infant's unobservable cognitive schema. At a high level, a formalized linking hypothesis corresponds to an input-output map. The inputs are pairs consisting of a) a cognitive schema (i.e. the representations or expectations acquired during habituation) and b) one or more test stimuli. The outputs are predictions about the infant's visual fixation behavior in response to each test stimulus.

There are several ways to formalize a linking hypothesis, and we distinguish between two general approaches. The first is a quantitative approach: given a cognitive schema and a test stimulus, a quantitative linking hypothesis outputs a numerical prediction of the infant's fixation time on that test stimulus. We can compute this numerical prediction using a formal notion of complexity (e.g. the *information content* of the test stimulus, or a measure of logical coherence between test stimulus and inferred representation) or expectedness (e.g. the posterior likelihood of the test stimulus given habituation stimuli and prior expectations). We can then generate a numerical prediction of fixation time as a function of this complexity measure. The exact details of this computation depend on a) how we formalize the hypothesis space, and b) the nature

of the linking hypothesis we are formalizing.[40]

A second approach is more qualitative: given a cognitive schema and two or more test stimuli, we again compute the same complexity measure for each test stimulus as above. However, rather than mapping these values to numerical fixation time predictions, we consider the ratio of these values to predict which test stimulus the infant will fixate on for longer. If test stimulus s_1 is significantly more complex or unexpected than s_2, we can predict that the infant will fixate on s_1 for longer, without having to explicitly predict the fixation time itself. This method is generally easier to apply, as it requires no numerical calibration or estimation to match observed fixation times.

To summarize this methodological framework at a conceptual level: we represent the infant's framework theory using the Level 3 representations defined in chapter 3.5, and define our question of interest in terms of subsets of possible framework theories. We simulate the habituation phase using Level 2 inference, which specifies the particular cognitive schema (in the context of ToM tasks, the actor model) that an infant with a given theory will infer during habituation to a particular stimulus. We then use our formalized linking hypothesis to connect the inferred cognitive schema to a prediction about the infant's visual fixation response to the test stimuli. Putting these components together allows us to derive statements of the form: "an infant with framework theory T will infer actor model M from habituation stimulus S, which will lead to visual fixation behavior v in response to test stimuli t_1 and t_2." We can then invert these statements to make inferences about an infant's framework theory, based on their observed visual

[40]If, for example, we are formalizing a dual-process linking hypothesis, then the predicted fixation time would be a function of stimulus complexity *and* a sensitization rate, the latter being estimated through some other process

fixation behavior in a particular habituation task.

4.3.2 Technical details

To illustrate the details of the conceptual approach described above, we will model the habituation experiment performed in Woodward (1998).

Defining a hypothesis space of framework theories The first step is to define a set of candidate framework theories for the class of stimuli used in these experiments. For simplicity of presentation, we shall assume that each framework theory consists of a single structural model and a prior distribution over the induced parameter space. We shall encode the salient observable features of each stimulus using three variables $S = (s_0, a, s_1)$, where

- s_0 denotes the initial state of the stimulus (i.e. the position of the two toys, the position of the actor's arm and hand, etc.),
- a denotes the action (i.e. spatiotemporal profile of the actor's arm), and
- s_1 denotes the outcome, which encodes the same information as the initial state, but at the end of the trial.

The structural part of a framework theory for this class of stimuli consists of a set of variables containing (s_0, a, s_1), as well as any latent variables posited by the observer. For the purpose of this illustration, we restrict the latent variables to at most one hidden feature g and up to one bias parameter for each of a, s_1, and g. The structural model is fully determined by its variable set and a dependency relation over those variables, and we can interpret each structural model as a framework theory about how the relevant

class of stimuli are generated. Figure 4.3.1 illustrates four possible structural models consistent with these definitions.

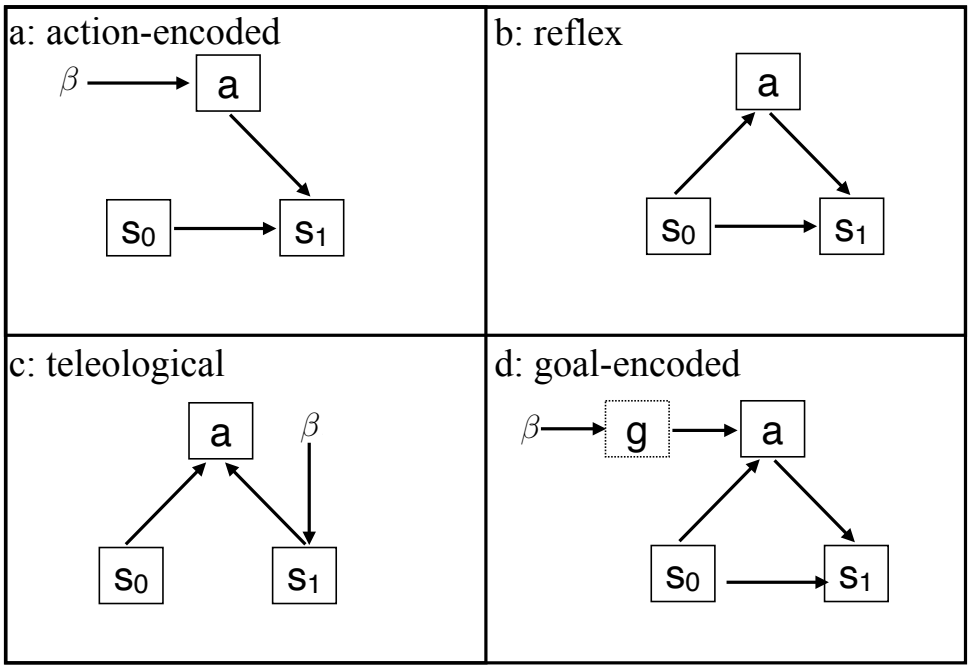


Figure 4.3.1: Examples of structural models for Woodward (1998) stimuli

Model 4.3.1a corresponds to an action-encoded model, according to which the actor has a bias towards certain arm motions, and the outcome results from the actor's chosen action. Model 4.3.1b corresponds to a reflex model, under which the action is a direct response to the initial configuration of the environment. Model 4.3.1c is an outcome-encoded or "teleological" model (Csibra & Gergely 1998), according to which the actor has a bias towards achieving a certain outcome, and selects the action that achieves that outcome.[41] Model 4.3.1d is a goal-encoded model, which is similar to the

[41]Note that this model contains an arrow $s_1 \rightarrow a$ which appears to go backwards in time. Thus, an observer with this model either a) does not recognize that causal relations

outcome-model, but represents the actor's goal as a hidden state that precedes the action, rather than the physical outcome which follows the action. Each of these models corresponds to one possible framework theory of reaching actions. Each allowable parameterization of a model corresponds to one particular representation (actor model) that an observer with that theory may consider for a single actor's reaching behavior. We can therefore define our hypothesis space of initial theories as the set of structural models over this variable set (and their corresponding parameter spaces) consistent with any constraints we derive from the subject's presumed background knowledge. The constraints we impose may be structural (e.g. eliminating any structural model with an arrow that goes backwards in time) or parametric (e.g. restricting the parameter $P(s_1|a, s_0)$ so that an actor who performs a short reach cannot grasp the toy that is further away).

With a hypothesis space formalized in this fashion, we can more precisely characterize the sort of qualitative claims about infants' representations that are tested in habituation experiments. The Woodward (1998) experiments, for example, explore whether infants encode reaching events in terms of the actor's arm motion or the target object of the reach. Given a hypothesis space of framework theories as described above, we can interpret each of these possibilities in terms of the structural model. In particular, if a model includes one or more trainable parameters corresponding to the a feature and no trainable parameters corresponding to the goal or outcome feature (e.g. figures 4.3.1a and 4.3.1b), then the infant would have to attend primarily to the arm-motion in order to infer the values of these parameters. We can therefore identify

---

cannot go backwards in time or b) interprets this arrow as a distinct type of non-causal relation.

these models as "encoding the reach in terms of the arm-motion." Similarly, if the structural model includes a trainable parameter corresponding to the goal or outcome (e.g. figures 4.3.1c and 4.3.1d), then the infant would have to attend to these features in order to train the parameters. We can therefore identify these models as "encoding the reach in terms of the outcome." In section 4.4, we derive a more exhaustive hypothesis space of structural models for the Woodward (1998) experiments and illustrate how to interpret this qualitative question as subsets of this space.

Modeling habituation In order to connect the infant's hypothesized framework theory with the cognitive schema they acquire during habituation, we need a way to model the process through which an infant acquires this cognitive schema. In the context of our computational framework, this corresponds to Level 2 inference: that is, given a framework theory T (i.e. structural model and parameter prior), we model habituation as the process of inferring an actor model M which best accounts for the behavior of the actor depicted in the habituation stimulus s. To this end, let s denote a habituation stimulus and S_n denote the evidence presented after the nth habituation trial (i.e. a sequence of n stimuli identical to s). A rational Bayesian observer interprets this evidence using Bayes' theorem:

$$P(M|S_n, T) \propto P(S_n|M)|P(M|T)$$

Here, $P(M|S_n)$ is the degree to which the observer believes in actor model M given evidence S_n and prior beliefs $P(M|T)$. As n increases, we can identify the increasing familiarity of the habituation stimulus with the increasing posterior likelihood $P(s|S_n)$

under the subject's inferred distribution. This is obtained by integrating $P(s|M)$ over all values of M (i.e. all representations compatible with the observer's initial theory), weighted by the posterior distribution $P(M|S_n)$.

If necessary, we can formulate explicit habituation criteria in terms of the posterior likelihood. If our criterion is reached after the nth habituation trial, we then simulate the observer's performance in the test phase by computing the likelihood of each test stimulus under the posterior distribution $P(M|S_n)$. This formalizes the notion that an infant interprets the test stimuli with respect to the representation inferred during habituation. By computing the likelihoods of the two test stimuli under this representation, we can apply our linking hypothesis to predict how the observer will allocate their visual attention to these stimuli.

However, we can also abstract away from methodological concerns regarding habituation criteria by simulating the observer's performance in the test phase after each habituation trial. Unlike a real-world habituation experiment, we do not have to wait for the observer to reach a pre-defined threshold before applying the above computation. We can therefore obtain simulated curves plotting the degree to which each test stimulus *would be* unexpected to an observer habituated to n habituation stimuli, for any value of n[42]. This allows us to generate simulated habituation curves as well as simulated plots of the observer's performance in the test phase after each habituation trial.

Actor models and fixation times The last step is to formalize the linking hypothesis that connects the infant's inferred actor model to a prediction about the infant's visual fixation behavior during the test phase. In the context of our

[42]If $n = 0$, this simulates the observer acting on prior expectations alone

computational framework, this corresponds to a form of Level 1 inference. Recall the general functional form of a formalized linking hypothesis: given a representation inferred during habituation and a pair of test stimuli, the linking hypothesis outputs a prediction about the subject's behavior in response to the test observations. This prediction can be quantitative (e.g. a numerical prediction of fixation time for each test stimulus) or qualitative (e.g. predicting which test stimulus the subject will fixate on for longer). The common assumption underlying most habituation experiments is that an infant will fixate longer on a stimulus that is more complex, unexpected, or novel, given the expectations acquired during habituation. The Bayesian framework provides a natural analogue of these notions in the form of the likelihood function.

For a given actor model M and test stimulus t, the likelihood term $P(t|M)$ is the probability of observing t given that M is true. If $P(t|M)$ is very low, this means that the test stimulus is highly unexpected, given the expectations entailed by M. Conversely, a high $P(t|M)$ indicates that t is largely expected by an observer who has inferred actor model M. We can leverage this interpretation of posterior likelihood to formalize both qualitative and quantitative linking hypotheses. For a qualitative linking hypothesis, suppose an observer infers actor model M during habituation and is then shown test stimuli t_1 and t_2. If $P(t_1|M)/P(t_2|M)$ is significantly lower than 1, we predict that the observer will fixate on t_1 for significantly longer than t_2, and visa versa if $P(t_1|M)/P(t_2|M)$ is significantly larger than 1. A similar approach is taken in Kemp & Xu (2009) to connect a generative model of object trajectories with predictions about infants' relative fixation times in object perception experiments.

A more quantitative approach is to use the posterior likelihood to compute an objective measure of stimulus complexity, given the observer's inferred representation.

This is the approach taken in Kidd et al (2012) to test the linking hypothesis that infants allocate their visual attention to stimuli that are neither too simple nor too complex. To this end, the authors define a generative model M of their experimental stimuli, and equate the complexity of a stimulus t with its negative log-probability ($-log(P(t|M))$) under M. This value- the *surprisal* of M- is often used in statistics and information theory as a proxy for information content, and quantifies the memory cost of encoding that stimulus for an ideal observer with representation M. We can therefore use the surprisal of a stimulus under a given representation as a basis for quantitative predictions about an infant's fixation time.

There are, of course, several different theories of habituation, some of which involve other factors in addition to stimulus complexity. Therefore, the precise way in which we connect the likelihood function to predictions about fixation behavior may vary depending on which linking hypothesis we apply. However, nearly every account involves, in some way, a notion of stimulus complexity or unexpectedness. While there may be additional technical differences for different implementations, the likelihood function provides a natural basis for computing the complexity (or unexpectedness) of a stimulus under a given representation.

Lastly, note that there are two different ways that one can use this reasoning to test hypotheses about infant habituation. The first is to apply a fixed linking hypothesis (formalized as some function of posterior likelihood) to a set of generative models, to determine which models induce fixation behavior consistent with infant behavior. This is useful for testing hypotheses about how infants represent stimuli, similar to Kemp & Xu (2009). The second is to assume a fixed generative model of a class of stimuli, and generate looking time predictions under multiple complexity-dependent linking

hypotheses. This is applicable for testing the linking hypothesis itself, similar to Kidd et al (2012). For our present purposes, we shall focus on the first application for the rest of this paper.

4.3.3 Empirical interpretations of the Bayesian framework

In recent years, rationalist approaches have become increasingly common in cognitive science, most frequently using the formal machinery of Bayesian inference. There are, however, different perspectives regarding how we ought to interpret such models, and how they may be useful for studying cognition (for a more thorough review of these perspectives, see Chater et al 2008, Griffiths et al 2008, Jones & Love 2011, Lee 2011). In order to understand these perspectives, it is important to understand the so-called "three levels of analysis" that are typically identified in cognitive modeling (Marr 1982). The first is the *computational* level, where we characterize the abstract problem that is solved by some cognitive function and the information involved in that problem. At the *algorithmic* level, we characterize the process through which a cognitive agent might solve that problem, including the representations involved and how those representations are manipulated. At the *implementation* level, we identify how these representations and algorithms might be implemented in the relevant physiological substrates. Of course, these levels are not completely independent; knowledge at lower levels of analysis can provide constraints on hypotheses at higher levels.

Many papers that take a rationalist approach to cognitive science restrict their interpretation to the computational level of analysis; they present the model as useful way of characterizing the cognitive problems being solved, rather than a literal claim about the processes through which they are solved. Other papers adopt a more realist

perspective, treating the model as a hypothesis about the cognitive representations involved in such processes. Our perspective for the methodological framework in this chapter is similar in spirit to the latter, though somewhat different in application. Much of the work in explaining human cognition with generative models focuses on "existence demonstrations;" that is, demonstrating a certain generative model which, when appropriately parameterized, approximately replicates human performance in some cognitive task (e.g. categorizing novel objects, learning novel words, etc.). Our framework is similar in that we interpret the model as a candidate hypothesis about how infants represent stimuli. However, rather than identifying individual models which approximately replicate infant behavior, our approach is to characterize a broader space of possible models of a class of stimuli, and interpret qualitative claims regarding infant knowledge as subsets of these models. This, we argue, provides a more precise way of specifying qualitative hypotheses about infants' knowledge, and can assist us both in answering questions and identifying more useful (and tractable) questions to ask.

Our interpretation and use of rationalist assumptions in this chapter is somewhat similar to the use of utility functions in economics and decision sciences. In particular, economists are interested in people's preferences, and how they act on those preferences to make economic decisions. Of course, we cannot directly observe a preference, nor can we directly observe the cognitive processes underlying decision making, so any attempt to study this subject requires some assumptions regarding what an agent's behavior reveals about their preferences. To this end, the overwhelmingly common approach is to model an economic decision maker as an approximately rational agent who optimizes some personal utility function. This assumption is flexible enough to capture nearly any pattern of decision making behavior, and provides economists and decision scientists

with a unified language in which to formulate hypotheses about agent preferences, and connect those hypotheses to predictions about agent behavior. We view the role of rationalist assumptions in empirical cognitive science similarly: we are interested in the knowledge and representations involved in infants' behavior, but we cannot observe them directly. We therefore assume that an infant's behavior reflects some approximately rational inference process over a representation (or space of representations) of the relevant stimuli. This provides a unified language for formulating hypotheses about infants' cognitive representations, and connecting those hypotheses to predictions about infant behavior.

4.4 Case study and simulations

Here, we illustrate our methodological framework in greater detail by simulating the Woodward (1998) experiments. We demonstrate how to construct a formalized hypothesis space of framework theories, interpret the experimenter's original question in terms of this hypothesis space, replicate the experiment via simulation, and use the results of these simulations to better analyze and interpret the results of the original experiment. Note that the main purpose of this initial demonstration is to validate our framework against existing data and a qualitative interpretation of that data. We discuss other potential applications of the framework more extensively in the next section.

4.4.1 Setting up the simulations

The first step is constructing a hypothesis space of candidate framework theories, which we briefly outlined in 4.3.2. This construction has three parts: first, we identify the

potentially relevant features, including observable stimulus features and latent features which may be posited by the observer. For observable features, we use our three-feature representation $S = (s_0, a, s_1)$ described in 4.3.2. For this illustration, we restrict the latent features to some subset of $\{g, \beta_a, \beta_g, \beta_{s_1}\}$, where g is a binary goal feature, and the remaining variables are bias parameters for a, g, and s_1, respectively. Second, we identify constraints on hypotheses based on what we can reasonably assume about the observer's background knowledge. Finally, we define our hypothesis space as the set of all structural models consistent with these constraints, along with each model's corresponding (possibly constrained) parameter space. This construction leaves us with 14 possible structural models and corresponding parameter spaces (see Appendix C1 for a full specification of constraints and models). These correspond to the 14 framework theories that we (the experimenters) consider as plausible candidates for the infant's theory of reaching actions. Our replication will therefore involve 14 sets of simulations, one for each candidate theory.

The next step is to define our question of interest as a subset of this hypothesis space. For this replication, our question of interest is: do infants encode reaching events in terms of the arm-motion or the outcome? To formalize this, we must identify subsets H_1 and H_2 which correspond to these two possibilities. This identification can be defined in terms of a model's dependency relation: an H_1 or "motion-encoded" model contains at least one of $s_0 \to a$ or $\beta_a \to a$ and cannot contain either $s_1 \to a$ or $g \to a$ (i.e. actions may directly depend on external circumstances and/or the actor's internal biases, but not on outcomes or goals). An H_2 or "outcome-encoded" model contains at least one of the arrows $s_1 \to a$ or $g \to a$ (i.e. actions directly depend on outcomes or goals).

To model habituation, we simulate an observer with framework theory T (consisting

of one of the 14 structural models and a uniform prior over the corresponding parameter space) inferring an actor model M from a single repeated instance s of an actor's reaching behavior (i.e. the habituation stimulus). For each simulation, we plot the observer's "habituation rate" by plotting, for each habituation trial, the posterior likelihood of the habituation stimulus according to the observer's posterior distribution (i.e. $P(s|S_n) = \int P(s|S_n, M)P(M|S_n, T)$).

To model the test phase, we apply a qualitative linking hypothesis: given the observer's posterior distribution $P(M|S_n)$ after the nth habituation trial, we predict that the observer will fixate on test stimulus t_1 longer than t_2 if and only if the posterior likelihood of t_1 is significantly lower than t_2 (i.e. $P(t_1|S_n) << P(t_2|S_n)$). This connects a hypothesis about the observer's framework theory with a prediction about the observer's relative fixation times during testing. Note that, if our goal were to replicate quantitative predictions or trends, we would need to perform a more rigorous parametric analysis and comparison against existing results. However, given the qualitative nature of the predictions we seek to replicate (preference for one stimulus over another), little analysis is needed to perform the current validation of our framework. Additionally, we can abstract away from methodological concerns about termination criteria by simulating the test phase after each habituation trial, rather than waiting until a termination criterion is reached. We plot the observer's predicted test-phase performance after each of a large but fixed number of habituation trials.[43]

[43]See appendix C2 for simulation specifications and C3 for examples of habituation and test curves

4.4.2 Results and analysis

Table 1 shows the compiled results from our 14 simulations (one for each framework theory). In each row, a preference for the "new-goal" event indicates that the posterior likelihood of the new-action event reaches at least 50% higher than the posterior likelihood of the new goal event (and visa versa for "new-action"). While we do not use explicitly coded habituation criteria (as we can simulate test results after any number of habituation trials), the posterior likelihood consistently reached a 150% threshold of initial likelihood after 6-9 trials across all simulations.

Table 1: Simulation results

Model	Hypothesis	Preference
h_1^1	H_1	New-action
h_1^2	H_1	None
h_1^3	H_1	None
h_1^4	H_1	None
h_2^1	H_2	New-goal
h_2^2	H_2	None
h_2^3	H_2	None
h_2^4	H_2	New-goal
h_2^5	H_2	None
h_2^6	H_2	New-goal
h_2^7	H_2	New-goal
h_2^8	H_2	New-goal
h_2^9	H_2	None
h_2^{10}	H_2	None

Based on these simulations, only one model (h_1^1) regards "new-action" as more unexpected, while five models ($h_2^1, h_2^4, h_2^6, h_2^7$, and h_2^8) regard "new-goal" as more unexpected. This allows us to begin assessing the question posed in the original experiment: do infants encode reaching actions in terms of the arm movement or the

target object? Based on this table, we see that every model which results in a preference for the new goal event belongs to H_2. This validates the experimenter's assumption that an observer who attends longer to the new-goal test event encodes the habituation event in terms of its outcome. Similarly, the only model which results in a preference for the new-action event belongs to H_1.

In addition to this initial validation, these simulations help us address some of the concerns raised in section 4.2. First, we noted the difficulty of drawing conclusions about an infant's internal representations when the habituation and test stimuli differ along inferred features, such as the actor's goal. While the looking time data alone tell us which stimuli appear more unexpected to the infant, this does not directly tell us how the infant represents the stimulus, especially when some of the relevant features are not directly observable. To this end, replicating an experiment in this framework helps us determine what distinctions we can and cannot infer among candidate representations, based on the data generated from that experiment.

In this case, we can rule out an action-encoded (H_1) representation (and a subset of outcome-encoded or H_2 representations) for any infant who attends significantly longer to "new-goal" Note, however, that among the models which develop a preference for "new-goal," one model (h_2^1) does not involve a latent goal variable. Rather, model h_2^1 identifies the goal with the physical outcome that follows the action (variable s_1), and predicts the action as a function of the initial state and target outcome. This corresponds to a *teleological* model of intentional actions (Csibra & Gergely 1998), which explains actions by relating them to physical constraints and outcomes through a principle of rationality. A teleological model is often distinguished from a *causal-mentalistic* model, which applies a similar principle of rationality, but involves a

hidden mental state attributed to the actor. Under a causal-mentalistic model, the actor's goal is a distinct latent variable which *causes* the action (and therefore precedes it temporally). Under a teleological model, the actor's goal is the literal physical outcome which *explains or justifies* the action. The results of these simulations demonstrate that the Woodward (1998) experiments cannot distinguish between these two possibilities. In order to make this distinction, one would need stimuli in which the actor's goal differs in some way from the physical outcome that follows (for example, see Brandone & Wellman 2009). Thus, replicating the experiment in this framework helps us clarify what inferences about infants' internal representations are justified by a given experiment, and what kind of experiments we need in order to make certain inferences.

A second challenge is distinguishing the expectations an infant acquires during habituation from the expectations the infant acquired prior to the experiment. This framework helps us separate these two kinds of expectations. To this end, note that our simulations were performed with uniform prior distributions over all trainable parameters. Intuitively, this encodes an assumption that the observer has no prior beliefs or expectations regarding the actor's biases. The results of these simulations solely reflect the expectations the observer acquired during habituation, and therefore illustrate the baseline expectations an observer would form in the absence of any prior expectations. However, we can further determine the influence that an observer's prior beliefs would have by simply changing the prior distributions from which trainable parameters are drawn. We can therefore use our framework to generate predictions for observers with specific structural models, i.e. predictions of the form "if an observer's framework theory T consists of structural model $\mathcal{M}$ and prior expectations P, they should prefer stimulus x over y after habituation to z." By holding $\mathcal{M}$ fixed and varying

P, we can predict the observer's post habituation preferences for different stimuli. Conversely, by holding some hypothesized $\mathcal{M}$ fixed and *observing* a subject's post-habituation preferences, we can infer properties of P, the agent's expectations formed prior to the experiment. Thus, by replicating habituation experiments in our framework, we can help validate the reasoning underlying an experiment, clarify what distinctions among candidate representations we can and cannot infer (and what experiments we would need to infer a given distinction), and separate the influence of an observer's prior expectations from the expectations they acquired during habituation.

4.5 Conclusions and further applications

Because there are so few ways to obtain data relevant to infant cognition, most of our knowledge comes from fixation time experiments. There are, however, some serious concerns regarding their proper design and interpretation. As we have argued, there is a gap in the relevant literature at the cognitive level: there are regression analysis models for assessing practical questions of experimental design, and connectionist models for exploring the neurological substrates underlying habituation. But, in order understand how an infant represents a stimulus and what we can infer about this representation from habituation experiments, we need an explicit model of the representations in question. We believe that our methodological framework helps fill this gap by serving three main functions. First, it helps us more precisely formulate hypotheses about infants' cognitive representations, allowing us to interpret qualitative questions as sets of generative models. Second, it allows us to formalize linking hypotheses that depend on stimulus complexity or unexpectedness, and thereby connect hypotheses to behavioral

predictions. Finally, we can integrate these components to replicate and analyze fixation experiments via a simulated version of habituation. As we saw in our case study, this helps us determine what questions an experiment can answer, what inferences are justified from a particular body of data, and what prior expectations or knowledge may influence an infant's performance in the test phase. In future work, there are several applications of this framework to explore. While the case study in section 4.4 applied our framework retroactively to existing data, we can also use the framework more constructively: by formalizing a space of representations for given stimuli, and a linking hypothesis connecting each representation to a behavioral prediction, we can determine which representations can and cannot be distinguished through behavioral data before an experiment is performed. We can then invert this reasoning to design stimuli that will be most useful for answering a given question about infants' cognitive representations. In future work, we will explore the constructive potential of our framework for assisting with design of experimental stimuli. Additionally, while this paper focuses on evaluating hypotheses about infants' representations under fixed experimental assumptions, we can also use the framework to test the experimental assumptions themselves. We can, for example, fix a single observer model of a stimulus and generate behavioral predictions under multiple linking hypotheses. By comparing these predictions against human infants' responses to the same stimuli, we can assess which linking hypothesis best fits experimental data (similar to the approach in Kidd, Piantadosi, & Aslin 2012). Finally, we can perform robustness checks against variations in subjects' intrinsic expectations, by simulating a population of infants with a distribution of different prior expectations and comparing habituation performance at individual and aggregate levels. This is similar in application to the regression analysis framework outlined in Thomas &

Gilmore (2004). Thus, this methodological framework provides a rigorous way to draw inferences about a subject's cognitive representations from their behavioral responses, and can help address many of the methodological and theoretical challenges inherent to studying infant cognition in particular.

5 Learning a Theory of Mind

Now that we have a methodology for empirical applications of our computational framework, we turn our attention to theoretical applications. In particular, we explore how we can use these models to provide theoretical justification for, and demonstrate the plausibility of, a rational constructivist account of ToM development.

5.1 Defining the observer's problem

Here we characterize the data the observer encounters, the inference tasks the observer must perform over this data, and the constraints under which the observer operates during this inference.

5.1.1 Data

Our aim is to model inference problems similar to the ones we face in our everyday social experience. As we discussed in chapter 2, there is a wealth of experiments in developmental psychology that assess how humans at various stages of development reason about other agents, using carefully constructed instances of commonly occurring social reasoning tasks. We shall draw on some of these experimental designs for the general forms (though not necessarily contents) of the data, expressed in the MDP

notation we presented in chapter 3. For example, Hilary et al (2012) construct a series of animated videos to test joint belief-desire inference in children at different stages of development. In these videos, an animated rabbit explores an environment with at least one observable object, and at least one wall which potentially obscures more objects. Children are then asked explanatory or predictive questions about the video (e.g: "based on the video, does the Rabbit prefer cookies or muffins?"; "based on the video *so far*, how do you think it will end?"). An example of such a stimulus (rendered as a grid world MDP system) is shown in figure 5.1.1.

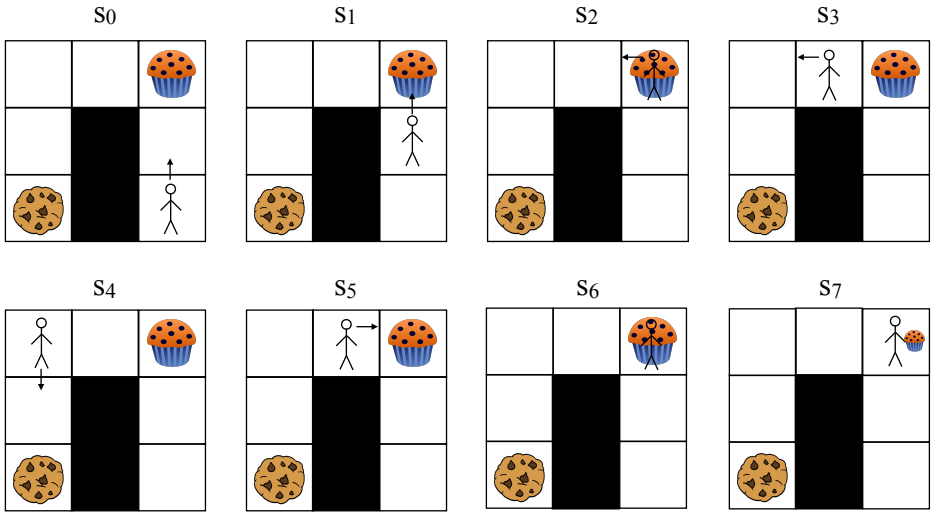


Figure 5.1.1: Example of a stimulus from a belief-desire inference experiment (Hilary et al 2012), rendered as a trial observation for a grid world MDP system. Black cells denote opaque walls. Arrow indicates direction of actor's line-of-sight

Importantly, many social inference tasks require reasoning about false or misrepresented beliefs, e.g. the classic "Sally-Anne" false belief task (and its many variants) that we described in chapter 2. To translate data of this sort into MDP form, we can modify the grid world system so that the grid layout itself may change during the

trial (possibly as a result of the observer's intervention, or another actor). An example of such a translation is shown in figure 5.1.2.

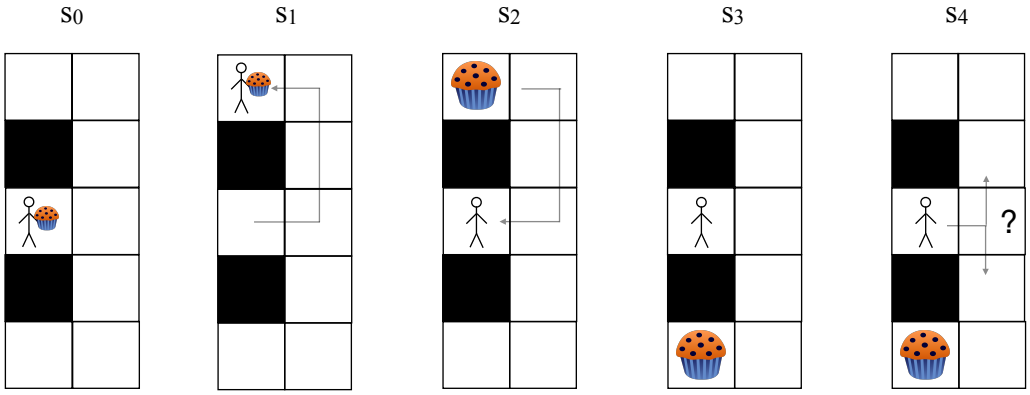


Figure 5.1.2: Grid world rendition of a "Sally-Anne" false belief task

For the scope of this dissertation, we shall make two simplifying assumptions when designing simulated data sets. First, we shall assume that, for a given actor, any two episodes of that actor's behavior are conditionally independent given the (correct) actor model.[44] Second, we assume that each actor's behavior is conditionally independent of each other actor's behavior, given the framework theory. This implies that the actors do not interact with or encounter each other during their own episodes (or if they do encounter other actors, those actors are not subjects of the observer's inference). Note that revoking these conditions would not pose a problem for the framework- rather, we make these assumptions to constrain the scope of the current project.

[44]Note that this is not the case if we want the observer to learn/reason about how the actor *learns*. While that is beyond the scope of the present dissertation, it can be modeled using same basic machinery

5.1.2 Inference tasks

For Level 1 tasks, we shall focus on behavior prediction and psychological explanation, which cover a wide range of (but do not exhaust) the social inference problems we face in real-life social experience. As with the simulated actor data, we will draw heavily from experiments in developmental psychology when designing the observer's Level 1 inference problems. For example, if the data depict a single grid world Sally-Anne trial, the observer may be required to either predict the actor's final action (where will Sally look for the muffin?) or provide a psychological explanation of the final action (what were Sally's psychological states, and how did they cause her to choose that action?). More generally, behavior prediction tasks can vary in terms of the trial features to be predicted, and the degree of accuracy required. For example, the observer may have to predict the actor's next action at each step, or simply predict the final state of the trial (based on a few initial observations), or predict both the end state *and* the actor's path to the end state (based on a single initial observation).

Higher level tasks are defined in terms of the observer's Level 1 performance. That is, given the data and the tasks the observer must perform, the Level 2 task is to infer a "good" actor model for the Level 1 tasks; i.e. a model which allows the observer to perform the necessary tasks as well as possible, subject to the observer's cognitive constraints (see next section). Similarly, given data from across multiple actors and a set of Level 1 inference tasks over that data, the observer's Level 3 task is to infer a "good" theory, i.e. a theory which entails "good" actor models.

5.1.3 Constraints

We shall consider three kinds of factors which may constrain the observer's inference. The first kind are computational constraints: that is, what are the observer's computational resources? To this end, we shall assume that the observer can construct and manipulate generative models in a functional PPL with marginalization operator and conditioning operator. While this may seem like a strong assumption, functional probabilistic programs are in fact a very general representation system, and can be used to realize a number of popular cognitive modeling frameworks, including Causal Graphical Models (Chater & Oaksford 2012) and Neural Networks (Goodman et al 2016).[45] In addition to the core components of a functional PPL (stochastic primitives, marginalization, conditioning, and recursive composition), we assume that the observer may use the following in constructing programs:

- An accurate physics model for each MDP system in the data. This knowledge is encoded in each system's state-transition function and state-to-action function.[46]

- Some form of (discounted) utility-computation program, which computes the (discounted) value of a sequence of actions given a value function over states. In

[45]Note that there is a straightforward way to interpret the problem of training a neural network as an instance of Bayesian inference. In particular, if D is the data, W is the parameter space of the network, and L is the loss function, we can interpret the problem of training the network as maximizing the posterior probability $P(W|D, L)$. The prior distribution over parameters is determined by the regularizer

[46]Though a worthwhile extension of this framework is one in which the observer can learn about the environment by watching the actor's behavior

grid world examples we have used so far, in which goals are defined in terms of a single object or feature, this program reduces to the path-length program we used in our example models.

- The primitive functions associated with specific variable types, i.e. the reference function for symbolic "pointer" variables, the similarity function for representational variables, and the value function for valuative variables.

The second kind of factor we consider are simplicity constraints, which determine the observer's preference between models that fulfill Level 1 task goals equally well (i.e. pragmatically equivalent models). Simplicity is an important notion of philosophy of science, and is especially relevant in cases (like this) where there are many (often infinitely many) distinct models that "fit" the data equally well. There are a number of ways to define simplicity measures, but a standard approach in the context of Bayesian inference is to use the number of free parameters in a model as a proxy for simplicity (e.g. Bayesian Information Criterion or Akaike Information Criterion- Myung & Pitt 1997). In addition to being very straightforward to compute, this is often justified on the basis that, given two models which fit a data set equally well, the model with fewer parameters will have higher marginal likelihood than the model with more parameters (under certain simplifying assumptions). Beyond this computational-level Bayesian justification, a preference for models with fewer parameters fits our assumption that the observer has bounded memory & representational resources. Since the parameters of an actor model constitute the "trainable" part of the model, a model with fewer trainable parameters will require less information to be learned, stored, and recalled for each individual actor.

To this end, we can define this simplicity constraint as a requirement on the

observer's over-hypothesis $\mathbb{T} = P(T)$, where the framework theory T is itself a prior distribution $P(\mathcal{M})$ over actor models. In particular, we require that the over-hypothesis $P(T)$ should prefer theories which prefer simpler models. I.e. if $T_1 = P_1(\mathcal{M})$ and $T_2 = P_2(\mathcal{M})$ are two framework theories which support Level 1 tasks equally well, the observer should prefer the theory that assigns more of its probability mass to simpler actor models (i.e. fewer trainable parameters). In addition to the bias for fewer parameters, we add a secondary "weak simplicity" bias for models that involve fewer total distinct variables (we assume every model includes s_{i-1}, s_i, a_i, and at least one parameter). This preference is subordinate to the parameter bias, in the sense that if M_1 has fewer parameters than M_2, but M_2 has fewer variables than M_1, the observer should prefer M_1. But, if M_1 and M_2 have the same number of parameters, the observer should prefer the one with fewer variables. This reflects that the observer has bounded representational resources, and therefore prefers models that are less costly to represent. Intuitively, the number of variables corresponds to the "cost" of storing the model structure, and the number of parameters corresponds to the "cost" of storing the actor-specific information (for each actor). Thus, the cost of additional parameters increases proportionally with the number of actors being explained, while the cost of additional variables is independent of the number of actors.[47]

The final factor is what we refer to as the "bias for determinism." In particular, we shall require that the observer's framework theory T prefer actor models that allow for

[47]This is an oversimplification; the full cost depends on the representational and relational structure of variables, and the domain of each parameter. However, proper derivation of a single measure which balances all these factors is beyond the scope of this dissertation

more "deterministic" psychological explanations of behavior. Obviously, a fully deterministic explanation of actor behavior is generally unfeasible, so we need to clarify what we mean by a "more" deterministic explanation. To this end, recall that a psychological explanation of a trial observation d corresponds to a posterior distribution $P(H|d)$ over the actor's psychological states, given the trial observation (and model constraints). The degree to which $P(H|d)$ is more or less deterministic corresponds, intuitively, to the "peaked-ness" of the distribution, i.e. the degree to which it assigns most of its probability mass to a small number of possible configurations.[48] While exact measures of peaked-ness are a subject of some controversy in statistics (e.g. Westfall 2014), for the current project we can make do with a less precise notion. In particular, given two models $\mathcal{M}_1$ and $\mathcal{M}_2$ which fulfill Level 1 task demands equally well, the observer's theory should prefer the model which assigns more of its probability mass to *fewer* possible configurations of the psychological states.

A bias of this form plays an important role in keeping the observer's inference computations tractable. In particular, recall that Level 2 and Level 3 inference both involve marginalizing out an actor's mental states. That is, in order to compute the data likelihood $P(d|\mathcal{M})$ under a given actor model, the observer must expand this into the weighed sum

$$P(d|\mathcal{M}) = \sum_{h \in H_{\mathcal{M}}} P(d|h, \mathcal{M}) P(h|\mathcal{M})$$

This requires computing the prior probability of each possible configuration of the hidden states h, and the likelihood of the data under that configuration. The exact

[48]Note that a "maximally peaked" distribution is deterministic, i.e. a single point mass on one outcome

number of computations required depends on the structure of the model: if a model has several nested layers of hidden states, the marginalization computations may result in a combinatorial explosion of likelihood computations. The bias for determinism helps offset this in two ways. First, the fewer configurations of h which are allowable under $\mathcal{M}$, the fewer computations are required to compute these marginalizations (i.e. if the model only provides two possible psychological explanations of a particular instance of behavior, then this marginalization will only involve two components in the sum). Thus, a bias for determinism will result in more tractable marginalization computations. Second, an easy way to make these computations more efficient is to approximate the marginalization with a MaP estimate (or a sum over a handful of the most likely values), weighted by its MaP probability. In cases where this estimate is sufficiently accurate, it represents a significant decrease in computation time, as it only requires a single likelihood term to be computed. The bias for determinism therefore helps by encouraging models for which MaP estimates (or "n-most-likely" estimates) are more accurate. That is, distributions which are more deterministic (per our definition) can be more accurately approximated by a handful of the most likely values.

While it may be possible to derive a single measure that encompasses all three kinds of constraints,[49] that is beyond the scope of this dissertation. Instead, we will focus our analysis on trade-offs that occur between these dimensions and the Level 1 performance of the model.

[49] A promising direction for this problem is the "resource rationality" framework (Lieder & Griffiths 2020)

5.1.4 Plan for this chapter

Now that we have characterized the observer's inference problems, we shall spend the rest of the chapter looking at "rational" solutions to these problems, and how those solutions can be inferred. We will start by constructing a class of "minimal" actor models; that is, the most basic kind of model that can fulfill Level 1 task goals subject to the observer's constraints. We shall demonstrate that solutions which include "value-like" hidden states (i.e. a hidden state that shares certain functional and relational properties with our commonsense concepts of "goals," "desires," or "values"; we refer to such states as "G-states") are "more rational" (per our definition) than solutions without such states. This will require us to characterize the defining properties of G-states in a way that is compatible with both our intuitive folk psychology and our formal framework.

Once we have established this minimal actor model, we will consider what kind of data would drive the observer to revise or augment this model, and what kinds of augmentations are rational. We shall argue that the rational augmentations include hidden states that share certain functional and relational properties with our commonsense concepts of "beliefs," "perspective," or "awareness" (we refer to these as "B-states"). As above, we will have to characterize B-states in a way that is compatible with our intuitions and our formal framework. In doing so, we will demonstrate both a) a formal characterization of our commonsense psychological intuitions (i.e. possible representational structures, functional relations, and algorithmic realizations), and b) an explanation of how these commonsense intuitions are "rational" solutions to social inference problems.

This analysis will provide a rationalist justification for why a BDA folk psychology is

a "good" solution for social inference problems. However, it is not sufficient for understanding *how* and *why* we acquire this sort of folk psychology. To make that leap invokes an "ought implies is" fallacy: just because X is the solution we "ought" to adopt, doesn't mean that X is the solution that we will, in fact, adopt. To this end, we will draw on Level 3 of our framework to motivate an account of how this learning might actually take place. In particular, we will demonstrate how the domain-general inference mechanisms defined for Level 3 can leverage this normative notion of rationality for effective learning, and that the models which result from this learning process will generally have a BDA-like structure. In this sense, we will use this chapter to show not only that our BDA folk psychology is a rational solution to social inference problems, but that we may in fact acquire our BDA folk psychology *because* it is a rational solution.

5.2 Building a minimal actor model

5.2.1 Why the observer is not a behaviorist

What are the minimal components required for an actor model that grants the observer a reasonable degree of predictive accuracy over some agent's behavior, subject to the observer's constraints? Obviously this will depend on the exact structure and contents of the data, but we shall start with a minimal characterization: the data depict multiple episodes of some kind of structured[50] behavior, and there is no observable external force that drives this behavior.[51] Any model that can explain this data will require, at minimum

[50] i.e. not uniformly random behavior, in which case there is nothing to learn

[51] The ability to distinguish between animate objects/agents (which are capable of self-motivated action) and inanimate objects/non-agents (which are inert unless acted upon

1. some kind of (posited) internal motivating force that generates behavior (otherwise there is no behavior to predict), and

2. something to shape the behavior produced by this force (otherwise predictions have no structure)

A very basic model (i.e. fewest hidden states, minimal representational/functional complexity among hidden states) that provides these components is a "hidden-switch" model like the one in section 3.5.3. Recall that this model is defined by a binary latent state M_t, and a disposition parameter Θ. If M_t is "off," the actor is inert, otherwise the actor behaves according to their disposition θ_s for the current system state. In theory, the predictive probability distributions over actions induced by *any* actor model could be approximated arbitrarily well with the distribution induced by this hidden switch model. In fact, we don't even need the switch: rather than distinguishing between two (or n) "modes," we can define the parameter Θ so that, under certain state conditions $\mathcal{S}$ (the conditions in which the actor tends to be "off"), the disposition $\Theta(\mathcal{S})$ (i.e. action distribution) deterministically outputs "no action." We can therefore, in theory, define an actor model that does not attribute *any* hidden states, and simply attributes a *behavioral disposition* parameter Θ. We can think of Θ as a table of action distributions, one distribution for each possible system state.

Intuitively, a model of this form corresponds to a "behaviorist" folk psychology. That is, rather than explaining the actor's behavior in terms of posited psychological states, an explanation of this form *just is* a set of behavior distributions, parameterized by Θ. This disposition parameter is a function of the actor's past stimulus-response history and

externally) appears to develop in early infancy (Woodward et al 1993)

current state (shown in figure 5.2.1). A behaviorist framework theory is therefore a probability distribution $T(\Theta)$ over possible behavioral disposition parameters. Given a behaviorist framework theory T, the problem of inferring an actor model $\mathcal{M}$ from actor-specific data D corresponds to estimating the probability that the actor will take a particular action in particular state, given their stimulus-response history.

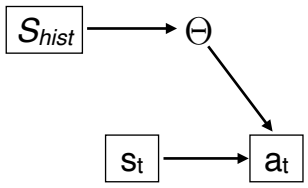


Figure 5.2.1: Graphical representation of a "behaviorist" actor model, which attributes a single (high-dimensional) "behavioral disposition" parameter and no hidden states

Given that models of this form can be parameterized to approximate any distribution over observable behavior, why is it not "rational" for our observer to be a behaviorist? Some would argue that an observer who adheres to traditional norms for rational scientific inference *should* adopt a behaviorist model, and that a model which posits additional cognitive structure is either more complex than necessary or inherently unscientific (depending on the extremity of one's views). B.F. Skinner presents a compelling case for these views in *Why I am not a cognitive psychologist* (1977), in which he argues that, because these posited cognitive states cannot be observed or intervened on, they cannot be the subjects of empirically verifiable hypotheses, and should therefore be of no concern to good scientists (or at least, good logical empiricists). So, given that our framework is built on a notion of "rational inference over empirical observations," why should our observer *not* be a behaviorist?

To answer this question, we must consider the constraints and biases we described in

section 5.1.3. In particular, we shall argue that

1. Cognitive actor models provide "simpler" explanations (fewer trainable parameters) than behaviorist actor models. As a pragmatic consequence, cognitive models can be trained more quickly from less data (for a particular actor).

2. While cognitive models include, by definition, more *psychological* variables than behaviorist models, behaviorist explanations often require more total variable information to predict future behavior.

3. Cognitive models provide more deterministic explanations of behavior than behaviorist models

Thus, we shall demonstrate that an observer with the biases and constraints we describe in 5.1.3 will generally have a strong bias for cognitive actor models over purely behaviorist actor models. To illustrate the first point, consider the "hallway" environment shown in figure 5.2.2.

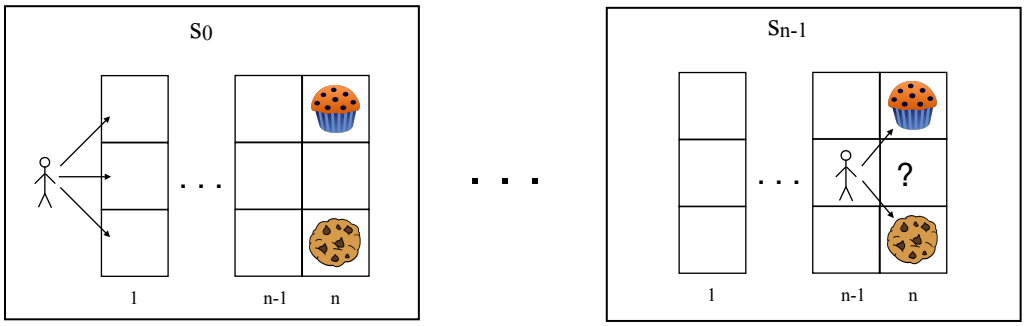


Figure 5.2.2: "Hallway" grid world environment, where n denotes the length of the hallway (number of columns in the grid). We assume that in each trial, the actor starts somewhere in the first column, moves right by one column in each step (up, down, or laterally), and that the trial ends when the actor reaches one of the two treats on the other end

Suppose the observer watches m trials $d_1, \ldots, d_m$ in variants of this hallway environment, all involving the same actor, and then begins to observe a new trial (involving that same actor) as it elapses. Suppose that the observer is asked to predict the actor's next action in this trial, given the initial segment $s_0, a_0, \ldots, s_t$. The behaviorist actor model is simply a parameter table $\Theta(s, hist_t)$, where s denotes the current system state, *hist* denotes the actor's behavior up to this point (in the current trial), and $\Theta(s, hist_t)$ encodes the probability that the actor will move (laterally, up, or down) in the next step. Thus, in order to do action prediction across these environments, the actor must learn, store, and recall a distinct parameter for each possible state/history configuration.[52]

Now suppose our observer has a simple cognitive actor model with a single hidden goal state, a value parameter β, and another parameter δ (e.g. error rate, mind-change probability, etc.). In section 3 we demonstrated several models of this form which would, if appropriately parameterized, allow for accurate behavior prediction in these hallway environments. For a given partial trial observation, the actor can predict the next action by a) inferring the goal state g (or the posterior distribution over g), then b) using this inferred value to compute the distribution over actions. The first computation involves sampling from a distribution with one parameter β (or two parameters if this is a "mind-change" model), and the second computation involves a deterministic computation (computing a shortest path to the goal) and a probabilistic computation involving another single parameter (error rate). These parameters are persistent across

[52]There are neural network techniques for learning behavior dispositions with fewer parameters than the number of possible state/history configurations, though these still require a significant number of trainable parameters relative to our cognitive actor models

episodes and can be used to do action prediction in multiple environments. Thus, a "cognitive" observer can make these predictions by inferring, storing, and recalling the values of as a few as two parameters, while a "behaviorist" observer requires a parameter for each state/history configuration. This also means that the behaviorist model will generally require significantly more data in order to infer parameter values that lead to accurate predictions. We illustrate this in a series of simulations, the results of which are shown in figure 5.2.3.

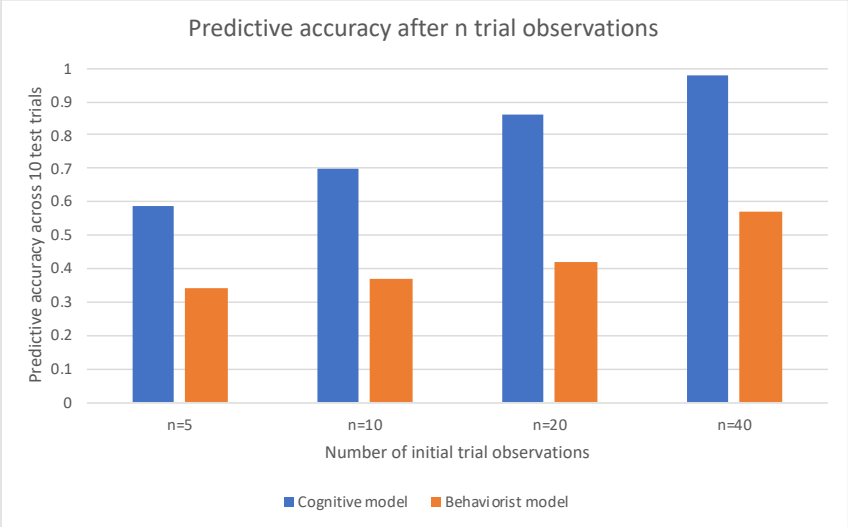


Figure 5.2.3: Results from a series of tests involving multiple hallway trials for a single actor. In each test, we simulate a behaviorist observer and a cognitive observer inferring an actor model over a data set of n trials (n=5, 10, 20, 40). We then generate 10 new trials with the same actor and have each observer predict the final action in each trial. Results reflect the predictive accuracy of each model (across the 10 test trials), given the number of observations in the data. As the results clearly show, the cognitive actor model yields more accurate predictions on fewer observations, demonstrating that the behaviorist actor model requires more data to train its (significantly higher number of) parameters.

In addition to the strong bias for fewer parameters, the observer also has a weak bias for models that require fewer total variables. In this sense, the behaviorist actor model

may seem to be weakly simpler, as it requires no hidden states to compute actions. However, a hidden state can, in fact, reduce the total number of distinct variables required to compute a future action in certain scenarios. This is due to the fact that posited hidden states can "screen off" the dependence of future actions on past actions, whereas the behaviorist model may require state/action information from arbitrarily far in the past in order to accurately predict a future action. To illustrate this, consider the partial hallway trial shown in figure 5.2.4. Given the system constraints, the actor must either move diagonally up (to the muffin) or down (to the cookie) for the last action. Suppose that the observer must predict this action.

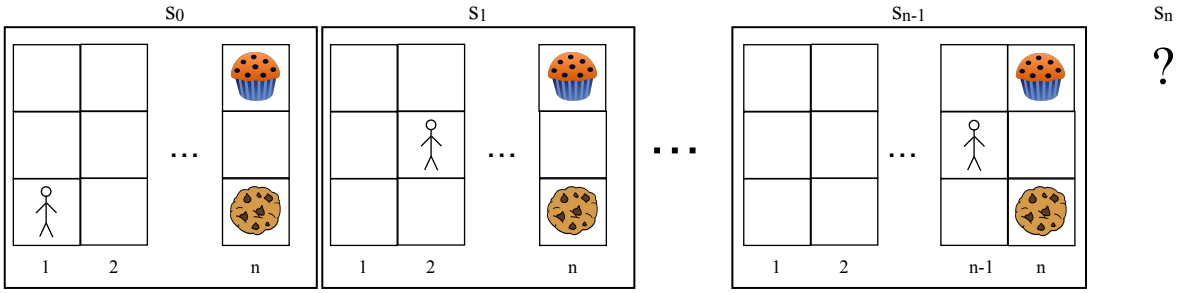


Figure 5.2.4: Example hallway trial

In order to make this prediction, a behaviorist observer would need to consider the action history for this trial up to now. Suppose, for example, that the actor started the trial in the bottom row, and moved diagonally up in one past step, while moving laterally (i.e. staying in the same row) for all other past steps. Intuitively, we should expect the observer to take their final step upwards towards the muffin; if their target had been the cookie, they would likely have stayed in the bottom row. In order for a behaviorist observer to make this inference, they would need to "rewind" the current episode to determine whether the actor started in the top, middle, or bottom row.

Depending on the length of the episode, a behaviorist observer may need to look arbitrarily far back in the past in order to make this inference.

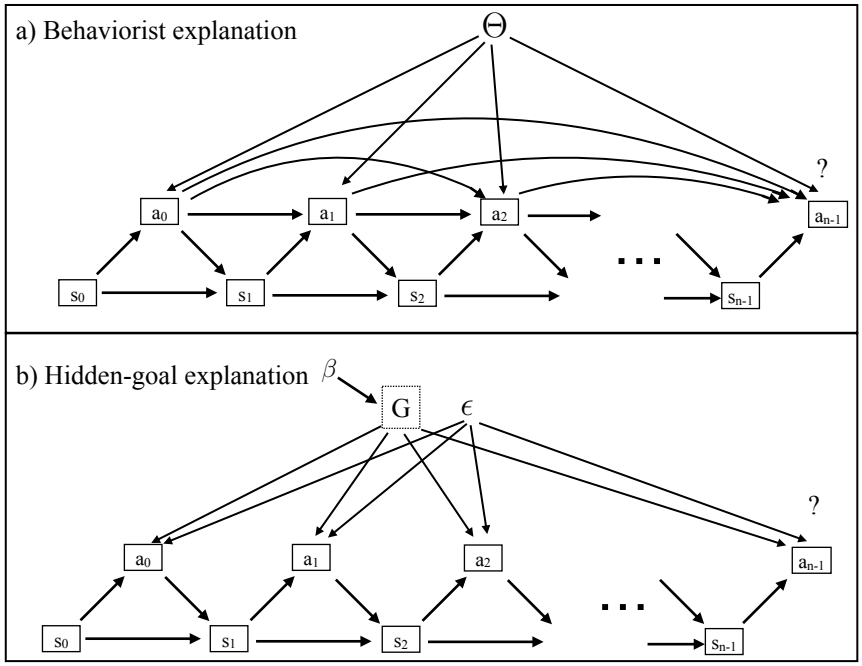


Figure 5.2.5: Psychological explanations of the trial in figure 5.2.4. Panel a) depicts the behaviorist's explanation, while panel b) illustrates a simple cognitive explanation. Note that, under the behaviorist explanation, each action is dependent on the current system state, parameter vector Θ, *and* each past action. That is, in order to determine whether the actor will move up or down in the final step, the behaviorist observer must recall whether the actor started in the bottom, middle, or top row. Under the hidden-goal explanation, however, the goal state g "screens off" this past information, allowing the observer to predict the final action without having to recall the actor's past actions

For a cognitive observer, however, this information is captured and maintained in the hidden state g. That is, as soon as the actor first moves up from the bottom row to the middle, the observer updates their estimate of g to reflect that the actor is most likely targeting the muffin, rather than the cookie. This hidden state is then maintained as persistent part of the observer's psychological explanation. When the observer reaches

the second-to-last state and needs to predict the final action, the relevant information is already present in the inferred hidden state, so the observer does not need to "rewind" the episode to make this prediction. Thus, the hidden goal state posited by the observer effectively "screens off" the past state/history information from the current prediction, thereby reducing the portion of the ongoing episode that the observer must "remember" in order to make predictions. This screening off effect can be more easily visualized by looking at the graphical representations corresponding to the inferred explanations (figure 5.2.5). Thus, even though the behaviorist model requires fewer *hidden* variables, it ultimately requires more observable state variables to predict future actions.

In addition to these two factors, the observer's bias for determinism encourages a cognitive, rather than behaviorist actor model. To illustrate this, consider a series of hallway trials (involving a single actor) in which the actor targets the muffin in 40% of trials and the cookie in 60%. Under the behaviorist model, this is explained in terms of a probabilistic behavior disposition: the actor's first step will be a move towards the cookie with probability .6, or a move towards the muffin with probability .4.[53] Their second action will be dependent on the first: in the first was the move towards the cookie, the second will most likely also be a move towards the cookie, and visa versa. Thus, by tracking the actor's sequential behavior within each trial, and comparing against past episodes in which the actor displayed similar behavior, the behaviorist observer explains the episode as a sequence of inter-dependent, random outcomes, with probabilities given by the actor's past stimulus response history. Under the cognitive model, however, the episode involves only a single random outcome, namely the actor's initial choice of goal at the start of the episode. Once the value of this hidden state is

[53]note that these two move sets are not mutually exclusive

fixed, the rest of the episode is explained nearly deterministically (up to noise/error) in terms of this single outcome. Thus, an observer with a strong bias for deterministic explanations will be skewed towards cognitive, rather than behaviorist actor models.

5.2.2 Building a minimal (cognitive) actor model

The previous section illustrates why, for an observer who is bounded by certain constraints and biases and faces certain inference problems, cognitive actor models (i.e. those that attribute hidden states) provide more rational solutions than behaviorist actor models (i.e. those that only attribute behavioral dispositions). Furthermore, in order to be useful for within-trial inference, these hidden states must track (in some way, to some degree) some observable feature of trial-specific data. So, what are the appropriate features to track, and what is the appropriate way to track these features in hidden states?

One important characteristic of episodic human behavior is its equifinal structure: relative to the number of possible outcomes that are feasible within a given environment, human behavior is organized around a small number of target outcomes. Importantly, this is persistent across environments: an agent with persistent preferences/values will regularly achieve the same distribution of outcomes across multiple environments (in which those outcomes are equally feasible). This creates a detectable pattern in the data which an observer can leverage to predict an actor's future behavior. Indeed, the equifinal structure of human behavior is thought to be an important perceptual cue in how infants first learn to recognize agents and predict agent behavior (Gergely et al 1995), and may also play an important role in how we learn to sequence continuous streams of behavioral data into discrete, episodic "chunks" (e.g. Buchsbaum et al 2012).

In order to leverage this equifinal structure for behavior prediction, the observer's model needs a posited hidden state that tracks, either implicitly or explicitly, some feature of the episode's end-state (or a counterfactually possible end-state). We refer to such states as "G-states," and shall argue that this category of hidden states shares important relational and representational features with our intuitive, folk psychological notions of "goals/values/desires/etc."[54] There are several ways that a G-state can encode this end-state information, either implicitly or explicitly. The "leanest" kind of G-state is simply a predicted final outcome for an episode (or some feature of that outcome). This corresponds to what is sometimes called a "teleological" action model (Csibra & Gergely 1998), which explains actions in relation to their outcomes and environmental constraints. Importantly, a teleological model is non-mentalistic, in the sense that the hidden state is not attributed to the actor directly, but to the trial itself (what *is* attributed to the actor is a tendency to achieve certain outcomes). An example of an actor model with this form of G-state is shown in figure 5.2.6(a).

The next "leanest" form of G-state consists of a pointer to the actor's intended outcome (or some feature of that outcome). This is distinct from the previous form in that this G-state is a psychological state directly attributed to the actor, whereas a teleological G-state is a prediction about the outcome of the trial, and does not correspond to an attributed psychological state.[55] An example of an actor model with this form of G-state is shown in figure 5.2.6(b). We can also define a "rich" version of

[54]Henceforth we shall use "values" to refer to this general folk-psychological category

[55]The practical importance of this distinction will become apparent in the section 5.2.4, when we consider trials in which there is a mismatch between intended outcome and actual outcome

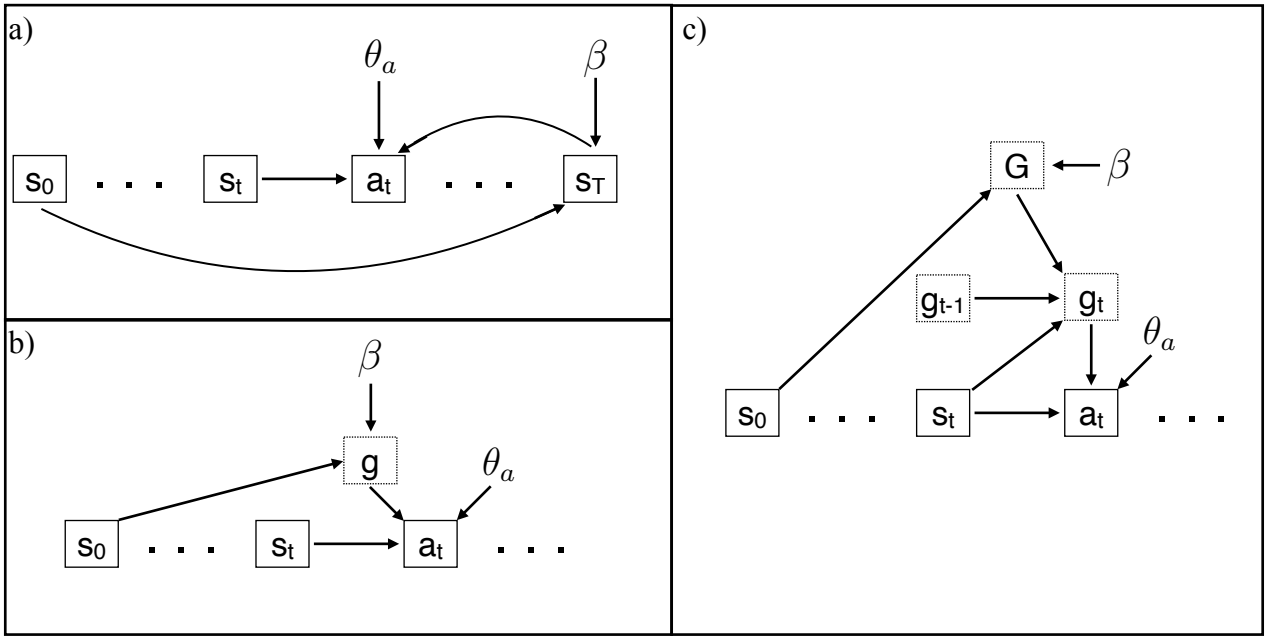


Figure 5.2.6: a) Example of a "teleological" actor model, which explains within-trial behavior in terms of a predicted future outcome, but does not directly attribute any psychological states to the actor. This model is parameterized by β, which captures the actor's tendency to achieve certain outcomes, and the action parameters θ_a. b) Example of a "simple desire" actor model, which explains within-trial behavior in terms of a hidden goal state, which "points to" the actor's intended outcome. c) A "sequential sub-goal" or "planning" model, which explains within-trial behavior in terms of a high-level goal G, which necessitates a sequence of sub-tasks g_{t-1}, g_t, etc.

this form, in which the G-state consists of a *representation* of an intended outcome (or some feature of that outcome), in conjunction with a program or similarity measure that assess the degree to which an actual state matches or satisfies this represented outcome. G-states may also be complex and hierarchically structured (e.g. Nakahashi et al 2016), such as a sequence of sub-tasks which serve some higher level goal (shown in figure 5.2.6(c)).

On a computational level, we can encode each of these forms as a value function over

states (or sequences of states, or sequences of actions). A simple predictor or pointer G-state can be encoded as a binary value function $v(s)$ which returns 1 if state s is the state being predicted/pointed to (or has the relevant feature). The value function for a representational G-state is defined by its similarity measure. A hierarchical G-state may be defined by a single value function which measures the degree to which the high-level goal is satisfied, and/or a sequence of value functions measuring the degree to which each sub-goal is satisfied. Conversely, any value function over system states can be interpreted as a G-state and used to construct an actor model. In reinforcement learning and control theory, it is common to characterize agents in terms of a value or reward function, and an action-selection "policy," and there are a number of general inference algorithms for learning an actor's value function (e.g. inverse reinforcement learning- Abbeel & Ng 2004) and predicting an actor's behavior from a general value function (e.g. Q learning- Strehl et al 2006). While this general form is computationally sound, for our purposes, it will be intuitively and notationally useful to distinguish between "sparse" G-states, which assign non-zero value to only one or a handful of possible system-states (e.g. a pointer or prediction), and "dense" G-states, which assign non-zero value to a large proportion of possible system-states. If the G-states are sufficiently sparse (which they will be for most of our examples), it it more convenient and conceptually clearer to define them in terms of the particular states or features they pick out, rather than specifying a full value function over states.

Now that we have defined G-states, and argued that any rational actor model must include some form of G-state, we shall consider how these definitions line up with the commonsense concepts of "values" (broadly construed) which play such a fundamental role in our psychological explanations. Recall in section 2 we distinguished between three

commonsense concepts that fall under this "value" category, which seem to loosely demarcate different stages in the child's developing understanding of actions. In particular, we distinguished between

- *drives*, which consist of an internally motivated urge or attitude and have no representational or referential content (e.g: "is thirsty"),
- *desires*, which consist of a target state/feature and a motivating attitude towards that state/feature (e.g. "wants a glass of water"), and
- *intentions*, which consist of one or more desires and a structured plan for resolving those desires, given the constraints of the surrounding environment (e.g. "is going to get a glass of water from the kitchen").

In the context of our definitions, we can define a "drive" as a value function that depends only on features specific to the actor's own bodily state. For example, we can encode "thirst" as a value function that places high value on any state in which the actor has had water, and low or zero value on any state in which the actor has not had any water. This captures the motivating attitude that corresponds to a drive, but is non-intentional in that it does not refer to or represent anything outside the actor itself. A "desire" can be realized using a pointer to/ representation of the target outcome/feature. Note that this means desires can be either representational or non-representational. Finally, we can realize an "intention" using a hierarchical G-state, which represents the high-level desire, as well as a particular plan (sequence of sub-goals) for realizing that desire in the current environment. Thus, we can interpret our commonsense concepts of drives, desires, and intentions as types of G-states. Though this does not mean that *every possible* G-state

will correspond to a particular commonsense notion of "value," the defining feature of G-states (that they encode persistent end-of trial information for the purpose of action prediction) is similarly fundamental to our commonsense psychological notions.

5.2.3 Reasoning with and inferring G-state models

Now that we have characterized the representational content and relational structure of G-states, we must characterize how G-states are used to explain and predict behavior. That is, how are actions determined by G-states (and system-states)? The fact that G-states encode information about a trial's (real or counterfactual) end-state imposes several intuitive constraints on the form of these relations. In particular, an actor with a particular G-state in a particular trial should be more likely to take actions which result in the target state (or actions that result in higher-valued states). However, this is not sufficient to predict the actor's intermediate behavior, i.e. behavior in states from which the target state is unattainable (by a single action). Intuitively, the end-state information encoded in a G-state is, by itself, only sufficient to predict some property of the (intended) final outcome of the trial, but not the path that the actor takes to that final outcome. Predicting this intermediate behavior requires an additional constraint on the model.

In human folk psychology, this constraint takes form of a "principle of rational action," which entails the intuitive rule than an actor will take the *best possible* action, given their goals and current physical constraints. In most cases, "best possible" is interpreted as "most efficient," so for many environments (e.g. our grid world environments), this principle reduces to "take the shortest path to the target

outcome."[56] While the explanatory and predictive advantages of such a principle are clear, it is less clear how we learn this principle. One possibility is that a principle of rational action is part of an innate ToM "module," which allows infants to perform certain inferences about intentional actions, and provides a conceptual "scaffolding" for the more mature theory of intentions that develops later in childhood. While there is strong evidence that young infants form expectations consistent with a principle of rational action (e.g. Gergely et al 1995, Philips & Wellman 2005), it is generally difficult to establish that a cognitive capacity is "innate," without either a) appealing to a strong "poverty of the stimulus" argument or b) demonstrating empirically that the capacity is clearly present from birth.

Due to the inherent challenges of infant cognitive studies (especially with extremely young infants, i.e. 0-3 months), it would be difficult to design a study that establishes knowledge of this principle in newborns, even if this knowledge is, in fact, present (in some form) at birth. While it is therefore difficult to establish that the principle of rationality is innate, it may be possible to give a poverty-of-the-stimulus *counterargument* to the effect that something like a principle of rationality *could* be learned, given that infants attend to the *outcomes* of episodic behavior (i.e. given that the observer posits a G-state). Intuitively, this would involve first learning a "soft" principle of rationality, corresponding to a general preference for shorter paths over longer ones.[57]

[56]For "dense" value functions, this is interpreted as maximizing the total discounted reward of the trial, and we can derive the "take the shortest path" interpretation as a special case of discounted utility maximization in which the utility function assigns non-zero reward to only one state

[57]Alternatively, infants may generalize a soft principle of rationality from self-experience.

For example, consider a framework theory which specifies a simple pointer G-state g and an action-selection function $af(s, g; \theta)$ with the following computational form:

$$P(af(s, g; \theta) = a) = \sigma_\theta(1/Length(shortestPath(T(s, a), g))$$

where σ_θ is a softmax function with scale parameter θ, i.e.

$$\sigma_\theta(x_i) = \frac{e^{\theta x_i}}{\sum_j e^{\theta x_j}}$$

Note that j ranges over the set of possible actions, so the denominator is a normalizing factor. Intuitively, the softmax function converts an unnormalized $n \times 1$ vector of reals into a probability vector. Thus, under an action function of this form, the probability of taking action a in state g is inversely dependent on the length of the shortest path to g for which a is the first step. The scale parameter θ determines the concentration of the induced distribution: if $\theta = 0$, then the resulting distribution is uniform, indicating no preference between shorter or longer paths. Larger values of θ induce distributions that are more heavily concentrated around higher input values. In the context of the framework theory defined above, an actor with a θ value of 0 is indifferent between shorter paths and longer paths, and is therefore equally likely to take any action that lies on a path to the target. An actor with a high ($>> 1$) value of θ will take the shortest

---

That is, an infant may recognize that *they* typically take direct paths to their goals, so by analogy other actors ought to do the same. While extrapolation from self-experience is beyond the scope of our current framework, it is an important augmentation to consider in the future

path near-deterministically. An actor with a large negative value of θ will prefer longer paths over shorter paths. This is illustrated graphically in figure 5.2.7.

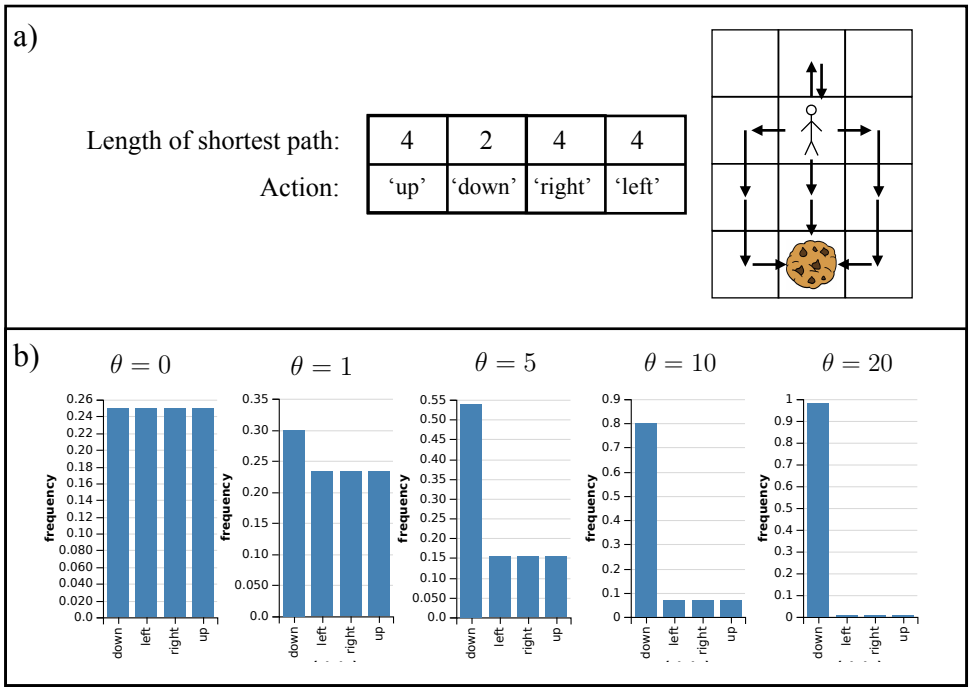


Figure 5.2.7: Example of the softmax action function defined above. Panel a) depicts a system state in which the actor's goal is the cookie. The table on the left indicates the length of the shortest path to the cookie which starts with the corresponding action. Panel b) depicts the actor's action distribution from this state, for varying values of the scale parameter θ. For $\theta = 0$, the actor is equally likely to take any action. For higher values of θ, the action distribution is more heavily concentrated on the shortest path. For high values of θ, the actor will choose the shortest path near-deterministically

A framework theory of this form can be specified as a prior distribution over its parameters $P(\theta, \beta)$, where β is a value parameter corresponding to G-state g. Given a prior distribution over θ, and a cross-actor data set, an observer could learn a "soft" principle of rationality by inferring a posterior distribution over θ. This is very similar to the example in section 3.5.2, in which an observer learns that actors tend to have peaked

or flat preferences, and can be performed with the same core program. In this case, the observer learns whether actors prefer longer or shorter paths (or have no preferences about path length). If the data depict actors who do, in fact, behave (mostly) rationally, the observer will learn this by inferring a posterior distribution over θ that is concentrated on high values of θ (strong preference for shorter paths). As the action distributions in figure 5.2.7 illustrate, a framework theory concentrated on sufficiently high values of θ will entail an "approximate" principle of rationality, i.e. the general principle that actors will near-deterministically choose the shortest path to a goal (unless some other factor intervenes). An example of this learning is shown in figure 5.2.8.

5.2.4 Developing and revising a G-state theory

In section 2, we reviewed experimental data showing that children develop an understanding of intentional behavior gradually, rather than all at once, and that this development appears to be loosely demarcated into several stages of understanding. Given that there are many possible G-state models, with different predictive and explanatory capacities, is it possible that these developmental stages reflect different underlying models, and that developmental transitions between these stages reflect inductive transitions in the child's higher level theory? To investigate this possibility, we shall consider the problem of "fulfilled vs. unfulfilled goal" understanding in infants (e.g. Brandone & Wellman 2009, Bellagamba & Tomasello 1999). Recall the stimuli forms used in Brandone & Wellman (2009), which extend an earlier design by Gergely et al (1995): in the "fulfilled goal" condition, an actor navigates around a wall to pick up an object on the other side. In the "unfulfilled goal" condition, the actor navigates around the wall, but fails their attempt to pick up the object. In the test stimulus, the wall is

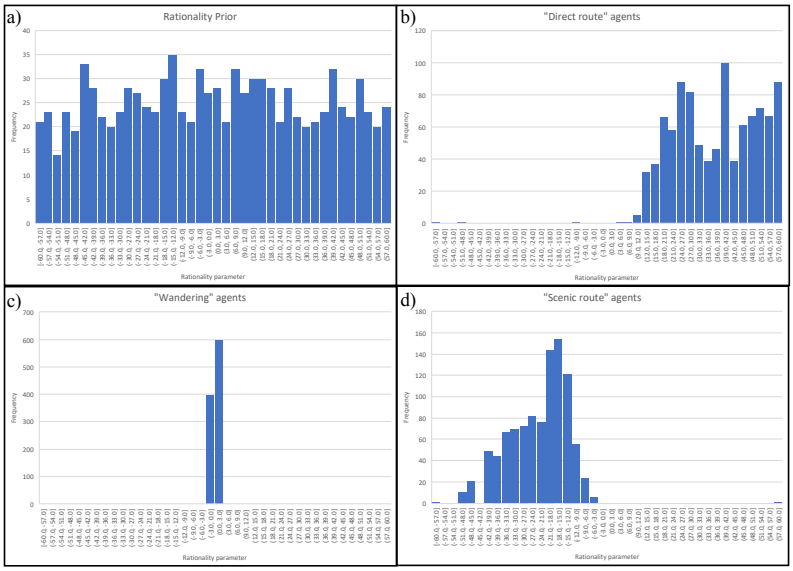


Figure 5.2.8: Inferring a principle of rationality. All distributions shown as histograms generated from 1000 samples. Panel a) depicts a uniform hyper-prior over θ, ranging from -60 to 60. Panel b) depicts the framework theory inferred for a population of agents who take the most direct path to the target object (up to 10% probability of error). As expected, the posterior is heavily skewed towards high positive values of θ. Thus, the observer has learned that these actors are nearly rational. Panel c) depicts the same results over a population of "wandering" agents, who are largely indifferent about the length of the path. This posterior is heavily concentrated around values of θ close to 0. Panel d) depicts the inferred theory for a population of agents who prefer the "scenic route." This distribution is skewed towards large negative values of θ

removed, and the experimenters test whether the infant expects the actor to take a direct path to the cookie (which was previously obstructed by a wall). Figure 5.2.9 shows a grid world rendition of these test stimuli.

All three age groups tested (10-, 14-, and 18- months old) consistently expected the actor to take a direct (rational) path when habituated to the "fulfilled goal" condition (which replicates and validates the results of Gergely et al 1995). However, only 14- and 18-month olds formed the same expectations from the "unfulfilled goal," condition; 10

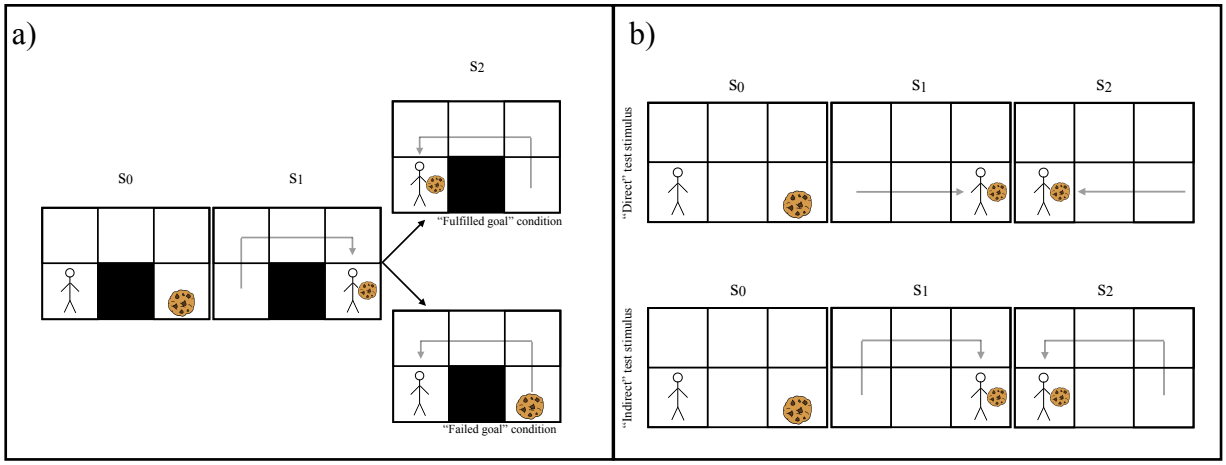


Figure 5.2.9: Grid world rendition of the habituation and test stimuli used by Brandone & Wellman (2009). In the habituation stimulus (panel a), the actor navigates around a wall to pick up an object, which they either pickup successfully ("fulfilled goal" case), or drop ("failed goal" case). In the test stimulus (panel b), the wall is removed, and the actor takes either a direct or indirect path to the object and successfully picks it up. In Brandone & Wellman, older infants (14- and 18-month olds) expected the actor to take a rational path to the cookie, regardless of whether they were habituated to the fulfilled or failed goal case. 10-month old infants who were habituated to the fulfilled goal case also expected rational behavior in the test stimulus (which is the same result produced by Gergely et al 1995), but those habituated to the failed goal case did not appear to form strong expectations about the actor's behavior in the test stimulus

month old infants who were habituated to the unfulfilled goal condition appeared to have no strong expectations about the actor's path in the test stimulus. One hypothesis as to what drives this apparent transition is that younger infants rely on a non-mentalistic "teleological" theory of intentional actions, which explains actions in terms of the outcomes they produce, while older infants possess a mentalistic "goal" theory, which explains actions in terms of a hidden psychological state that "points to" an intended outcome. We can flesh out and evaluate this hypothesis using a version of the "teleological" and "pointer" G-states described in the previous section.

To this end, we shall define two possible framework theories T_1 (teleological) and T_2 (goal-based). The first theory comprises two possible structural models: M_0, which constitutes an"irrational agent." Intuitively, M_0 is a simple behaviorist model which corresponds to the explanation that the actor's behavior is not a result of any deterministic internal states, and that the only way to form expectations about the actor is to tabulate the frequency of their actions in particular environments. The second model is a teleological G-state model like the one shown in figure 5.2.6(a). This model is parameterized by β, which captures the likelihood that a particular actor will achieve a particular outcome in a particular environment, and ϵ, which captures the probability that the actor will "drop" an item that they've attempted to pick up. Since the only hidden state in this model is the predicted final outcome s_T, the "mental update function" for this model simply samples a value of s_T from those outcomes that are attainable from initial state s_0, with probabilities determined by the preference parameter β. The action function $af(s_t, S_T, \epsilon)$ either a) deterministically outputs an optimal action towards the predicted final state S_T (if the actor is not currently in the target location), or b) attempts to pickup the target object from the current location, and successfully does so with probability $1 - \epsilon$.

The second theory is nearly identical to T_1, and consists of an "irrational" model and and "rational" model. The difference is in the rational model: instead of computing the action in terms of a predicted future outcome, the action is computed in terms of an *intended* outcome, which *points* to, but is not identical with, a possible future outcome. The intended outcome g is sampled at the start of the trial from among those attainable from initial state s_0, according to the same parameter β. The action is computed in the same way as before (with the same probability of dropping an object), but takes the

intended outcome g as input rather than the predicted outcome s_T.

Since each theory contains two models, and the corresponding models share the same parameter space, we can use the same set of parameter priors to define both theories. In particular, we shall assume that the prior distribution $P(M)$ between the two models within each theory is heavily skewed towards the rational model ($P(rational) = .9$), encoding the general knowledge that actors are much more likely to behave rationally than not. The prior distribution $P(\beta)$ is highly concentrated on probability vectors which are highly concentrated on "salient" outcomes/target states (i.e. those that involve reaching and picking up a particular object), and distribute a very small (but non-zero) amount of their probability mass on "non-salient" outcomes (i.e. any other possible system-state). This reflects the high-level knowledge that a previously unknown actor in a previously unknown environment will most likely target some object in that environment, but allows for the possibility of an "unusual" actor that targets other kinds of system states (e.g. someone who just likes to stand next to walls). The prior distribution over the error parameter $P(\epsilon)$ is assumed to be heavily skewed towards small values, encoding the general knowledge that actors aren't too clumsy, but will occasionally drop something.

Given these two framework theories T_1 and T_2, we shall argue the following:

1. An observer with a teleological theory (T_1), when presented with the habituation stimuli in figure 5.2.9, will form expectations about the test stimulus consistent with 10 month old infants in the Brandone & Wellman study (i.e. will expect rational behavior if habituated to the fulfilled goal case, and will have no strong expectations about the actor's behavior if habituated to the unfulfilled goal case)

2. An observer with a goal-based theory (T_2), when presented with the same data, will form expectations about the test stimulus consistent with older infants (i.e. will expected rational behavior in the test stimulus regardless of the habituation condition)

3. An observer with the constraints and simplicity biases we describe in section 5.1 will first prefer a T_1 theory, but will gradually come to prefer a T_2 theory when presented with cross-actor data that depicts a higher concentration of mismatches between intended and actual outcomes. That is, given cross-actor data depicting (approximately) rational goal-seeking behavior, a rational observer will first prefer T_1 over T_2, but will revise this preference as they begin to observe more intended/actual mismatches in the data.

To illustrate the first two points, we can model the habituation experiment described above as a two-stage inference problem, using the methodology we describe in chapter 4. In the first stage, the observer must infer an actor model for a novel actor, based on a single (repeated) episode of that actor's behavior. In the second stage, the observer must draw on this actor model to predict the actor's behavior in a pair of new (test) stimuli. We simulate this two-stage inference for two different observers, one with the teleological theory defined above (T_1) and one with the goal-based theory (T_1). The results of these simulations are shown in figure 5.2.10.

The results in 5.2.10 are divided into two columns- corresponding to the habituation condition (fulfilled-goal or unfulfilled-goal)- and two rows- corresponding to the observer's theory (T_1 or T_2)- creating a total of four cells. In each cell, we depict results from habituation-phase Level 2 inference problem (reporting the inferred posterior

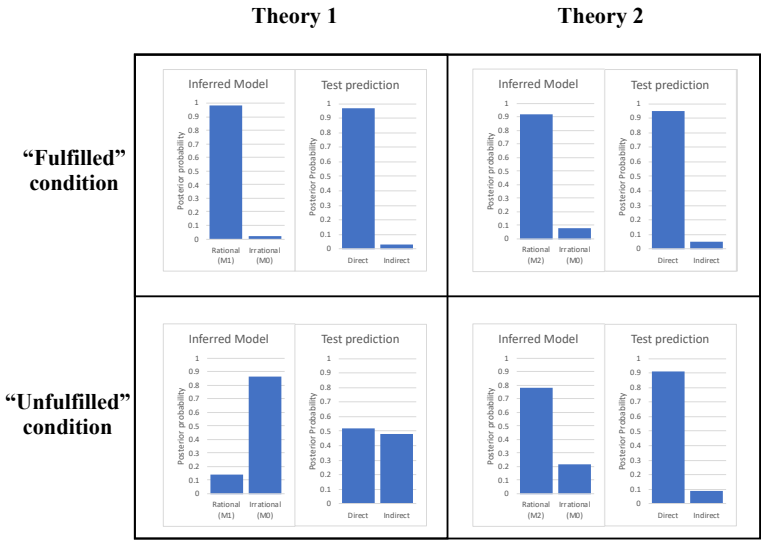


Figure 5.2.10: Results of simulated habituation experiment using a teleological theory and mentalistic theory. Columns indicate theories and rows indicate habituation conditions ("fulfilled" or "unfulfilled"). Each cell depicts two results. The first is the observer's posterior estimate that the actor is rational or irrational (according to that theory's model of rational behavior). The second is the observer's expectation for the test trial (i.e. probability that the actor will take a direct or indirect path). In the "fulfilled" condition: both a teleological observer (column 1, row 1) and a mentalistic observer (column 2, row 1) infer that the actor is very likely rational, and strongly expect the actor to take the direct path in the test trial. In the "unfulfilled" condition: the mentalistic observer (column 2, row 2) still infers that the actor is most likely rational, and still expects the actor to take a direct path, while the teleological observer (column 1, row 2) infers that the actor is most likely not rational, and forms no expectations about the actors test-phase behavior.

probability that the actor is rational, $P(rational|d, T)$), and the results of the subsequent Level 1 inference problem (reporting the posterior likelihood of each test stimulus, under the distribution induced by the inferred actor model).

In the fulfilled goal condition, both theories result in actor models with similar parameter values. In particular, both models infer a high probability that the actor is rational, both infer that the actor strongly prefers the one salient outcome (picking up

the cookie) to any other outcome, and both infer that the actor has a low error rate. Consequently, both actor models generate the same test-phase expectations, and place significantly higher posterior likelihood on the direct, rather than indirect, test stimulus. This is consistent with the expectations formed by infants in all three age groups in Brandone & Wellman (2009) for the fulfilled goal condition. In the unfulfilled goal condition, however, the inferred actor models are very different. Under the goal-based theory, the observer still infers a high probability that the actor is rational, but infers a different distribution over β and ϵ: in particular, β is skewed slightly (relative to the prior) towards a preference for "other" outcomes, and ϵ is skewed more towards higher error rates (relative to the prior). Intuitively, based on this single episode and the observer's (goal-based) framework theory, the observer has inferred that the actor likely intended to pickup the cookie but dropped it, *or* the observer *intended* to drop the cookie (though this explanation has lower posterior likelihood than the "mistake" explanation). Based on this inference, the observer still forms strong expectations that the actor will take a direct rather than indirect path in the test phase, consistent with 14- and 18-month old infants.

Under the teleological theory, the observer infers a high probability that the actor is irrational, and a low probability that the actor is rational, but strongly prefers to drop the cookie. Consequently, the observer forms no strong expectations about the actor's behavior in the test phase, resulting in similar posterior likelihoods for each test stimulus. This is consistent with 10-month old infants in the Brandone & Wellman experiments.

To understand why this occurs, we must look at the computations involved in Level 2 inference. In particular, let $T = \{\{M_1, M_2\}, P(M), P(\theta)\}$ be one of the two theories defined above (where $\theta = (\beta, \epsilon)$), and let $d = \{s_0, s_1, s_2\}$ denote the "unfulfilled goal"

habituation stimulus. The observer's problem in the habituation phase is to infer an actor model according to its posterior distribution

$$P(M, \theta | d, T) \propto P(d | M, \theta) P(M | T) P(\theta | M, T)$$

Here $P(M|T)$ denotes the prior probability that a new agent will have an irrational or rational model, and $P(\theta|M,T)$ denotes the parameter prior for each model. Since the irrational model has no parameters,[58] we assume this prior probability is 1 for that case.

As usual, we decompose the trial likelihood $P(d|\theta)$ into a product

$$P(d|\theta) = P(s_1|s_0, \theta) P(s_2|s_1, \theta)$$

and each term $P(s_i|s_{i-1}, \theta)$ is computed by marginalizing out the hidden state h. For this example, we distinguish between only two values of h: the one salient outcome C (holding the cookie), and O for "other." Thus, each term must be marginalized as

$$P(s_i|s_{i-1}, \theta) = P(s_i|s_{i-1}, h = C, \theta) P(h = C|\theta) + P(s_i|s_{i-1}, h = O, \theta) P(h = O|\theta)$$

Under the teleological theory, however, the hidden state h *just is* a prediction of the final

[58]Note that the irrational model will never produce coherent expectations for the test stimulus, as the irrational model simply tabulates the frequency of individual actions in individual environments. Since the test-stimulus takes place in a distinct environment, none of the frequency information captured during habituation transfers over to the test stimulus. For this reason, we can make the notation much more compact by treating the irrational model as if it has no trainable parameters

state s_2. This is a problem, as the final term in the trial likelihood is now

$$P(s_2|s_1, s_2 = C, \theta)P(s_2 = C|\theta) + P(s_2|s_1, s_2 = O, \theta)P(s_2 = O|\theta)$$

In this case, the likelihood function over the final state s_2 is conditioned on its own predicted value. Since $P(x|x = c) = 1$ for $x = c$ and 0 for any other value of x, this means that the term corresponding to the *intended but not realized* outcome (C) drops out of the likelihood computation. Intuitively, because the teleological model uses a predicted final state (as opposed to a counterfactual target state attributed to the actor) to form expectations about behavior, the teleological observer does not recognize the possibility that the actor intended one outcome but achieved another. Thus, the only two explanations the teleological observer can infer for the unfulfilled goal trial are a) the actor *intended* to pick up and then drop the cookie, or b) the actor is simply irrational. This demonstrates that a teleological model, which lacks an attributed psychological state, produces a similar pattern of expectations to those formed by 10-month old infants, but not 14- or 18-month old infants, in Brandone & Wellman (2009), and a simple goal-based model is consistent with 14- and 18- month old performance.

To address our third point, we must consider what would drive an inductive transition from a teleological theory to a goal-based theory, which requires Level 3 inference. To this end, consider an over-hypothesis $\mathbb{T}$ for a single-model framework theory, which specifies two possible model structures: a teleological model M_1, as shown in figure 5.2.6(a) (i.e. with a predicted final state in lieu of a psychological goal state), and a goal-based model M_2, as shown in figure 5.2.6(b) (i.e. with a posited psychological goal state). Since M_1 and M_2 share the same parameter space, a single framework

theory T consistent with this over-hypothesis consists of one of the two structures M, and a prior distribution $P(\theta)$ over the parameter space $\theta = (\beta, \epsilon)$. The over-hypothesis therefore specifies a prior distribution $P(M)$ over models, and a hyper-prior $P(P(\theta))$ for θ. In keeping with the simplicity biases described in section 5.1, we shall assume that $P(M_1) > P(M_2)$. This is due to the fact that both models require the same number and types of parameters, but only the goal-based model includes a posited psychological state. Thus, in the absence of any evidence, the observer should prefer M_1 over M_2.

Given this setup, we can simulate the observer inferring a framework theory using our Level 3 inference program. For simplicity of presentation, we set the program to only return the posterior distribution over model structures, and marginalize out the parameters. This allows us to visualize the posterior likelihood that the observer assigns to each of the two categories of theories (model structures), rather than the joint structure-parameter distribution. To construct the data for this experiment, we generate a cumulative sequence of 6 data sets (i.e. each d_i contains all the observations in d_{i-1}). Each data set depicts an equal number of trials for each of 5 actors. Each trial depicts an actor navigating (possibly around an obstacle) to an object, then attempting to pick it up, and possibly dropping it. In "fulfilled" trials, the actor may repeatedly pickup the object until they are successful. In "unfulfilled" trials, the trial ends after the actor's first attempt (i.e. if they drop the object, they cannot pick it back up). To construct the actor models, we draw β from one of two priors (flat preferences or peaked preferences), and ϵ from a $beta(.1, 1)$ distribution (skewed towards small error rates). We then apply our Level 3 inference program with the over-hypothesis described above. The results of this simulation are shown in figure 5.2.11.

The first two data sets depict only fulfilled trials, while each subsequent data set

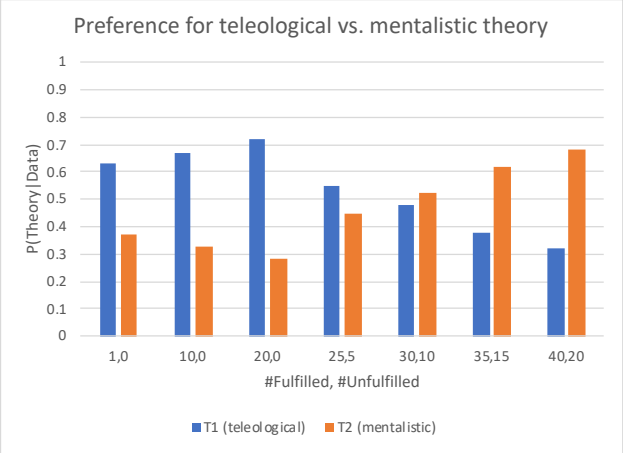


Figure 5.2.11: Each pair of columns depicts the observer's posterior degree of belief in a teleological (T1) or mentalistic (T2) framework theory, for seven different data sets. Each data set consists of an equal number of trials for each of 5 actors. In each trial, the actor navigates towards an object and picks it up, either successfully or unsuccessfully (i.e. drops the object and does not pick it up before the trial ends). The pair of numbers (x, y) below each pair of columns indicates indicates the number of successful vs. unsuccessful trials in that data set. We start with only successful trials, before introducing unsuccessful trials (the data sets are cumulative). The first panel $(1, 0)$ depicts the results after only a single trial, which is very close to the observer's over-hypothesis (i.e. prior preference between theories).

increases the concentration of unfulfilled trials. For the first two data sets, the observer's posterior likelihoods roughly correspond to the prior likelihoods. This is expected, as, both models can explain these data sets equally well, so the ratio of posteriors reduces to the ratio of priors. The reason for this is the nature of the data: even though actors have non-zero error rates (and may therefore make mistakes), the actors in these trials may continue acting until they reach their goal condition; e.g. an actor who drops an object may circle back to pick it up again. Since the teleological model explains the trial in terms of the *final* state, it can still accommodate mistakes that the actor can fix. As we introduce unfulfilled trials, however, the observer's preference for the teleological model

decreases, and they eventually come to prefer the goal-based model. This is due to the fact that, in cases where the actor is unable to realize their goal, the teleological model will always induce an explanation with low prior probability (namely, that the actor *intended* to stand in the same location as an object, but did not want to pick it up, but mistakenly picked it up and then put it back down). Even though both models can explain completed trials equally well, the fact that the teleological model struggles to explain failed-goal trials significantly reduces the data likelihood under that model, which shifts the posterior probability away from M_1 and towards M_2. This suggests that, initially, the observer will prefer the teleological theory, but as the observer is exposed to more fail-goal trials, they will gradually shift away from a teleological theory and towards a goal-based theory.

5.3 Adding beliefs

5.3.1 Why are G-states not enough?

As we saw in the previous section, G-state models, in conjunction with a principle of rational action/utility maximization, allow for a wide range of inference and prediction. Furthermore, G-states share important conceptual and structural features with our commonsense notion of "values" (broadly construed to include desires, goals, preferences, etc.). However, as we know from experience, values alone are often not sufficient to predict or explain human behavior. From the developmental data, it is unclear at what stage we first recognize this: on the one hand, classic False Belief tests (e.g. Wimmer & Penner 1983) clearly show that younger children (< 4 years) struggle to explain mistaken behavior in terms of false beliefs, a capacity clearly held by older children and adults.

However, there is strong evidence that younger children possess *some* understanding of how an actor's epistemic access constraints their behavior. Two- and three- year old children are clearly sensitive to an adult's visual access when formulating requests (e.g. will point out the location of a toy if the adult did not see it placed on a high shelf, otherwise will simply ask for the toy- O'Neill 1996). Even infants display expectations consistent with rudimentary false belief understanding (Onishi & Baillargeon 2005). This suggests that our capacity to track and reason about the epistemic states of other agents develops gradually, in conjunction with our capacity to track and reason about values.

On a theoretical level, it is worthwhile to ask why we cannot rely on G-states alone. To this end, consider a two-stage inference problem like the one the previous section: in the first phase, the observer must infer an actor model for a new actor from a single trial observation, then apply that actor model to explain and reason about a second episode. The two trials are shown in figure 5.3.1. For an observer with a mature BDA folk psychology, these two trials have a straightforward, (near)-deterministic explanation: in the first trial, the actor proceeds to the only treat they see (the cookie); once they reach the cookie, they see the muffin, which they prefer over the cookie; once they reach the muffin, they see the pie, which they prefer over the muffin. In the test trial, the actor can immediately see the pie, and moves directly towards it.

How would an observer with a G-state model explain the initial trial in figure 5.3.1(a)? This of course depends on the form of the model. In figure 5.3.2, we show the results of the two-stage inference process for each of three models. If the observer has a fixed-goal model (i.e. one in which the actor's goal is persistent throughout the episode), the only explanation of this trial is that the actor sought the pie over the cookie and the muffin, but made an unlikely sequence of errors resulting in a path significantly longer

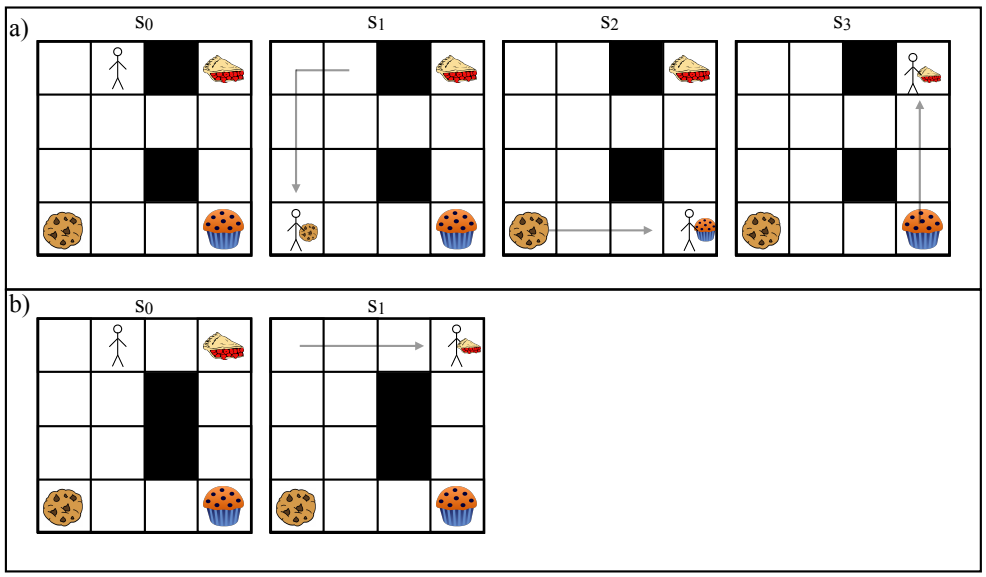


Figure 5.3.1: Two phase inference problem consisting of an initial trial observation (panel a) and a test trial observation (panel b).

than necessary. The inferred actor model for this observation will therefore have an unusually high error rate ϵ. Thus, under a fixed-goal model, the rational (MaP) explanation of the initial trial (one with highest posterior probability) attributes a high degree of randomness to the actor's behavior, conditioned on their mental state. This results in an actor model that provides less predictive power and more non-deterministic explanations, and assigns lower posterior probability to the test trial.

If the observer has a mind-change model (i.e. one in which the actor is strongly rational, but may change their mind during the episode), the explanation of this trial involves two mind-changes; the first may have taken place at any point between the actor's first movement to the left and their final movement to the cookie's location, the second may have taken place at any point between the actor's first movement toward the muffin and final movement to the muffin's location. As a result, a mind-change model

	Fixed goal model	Mind change model	Hierarchical goal model
Inferred parameter	0.45 (error rate)	0.8 (fickleness)	.02 (error rate)
MaP goal estimate for trial 1	M	[C,C,C,C, M,M,M,P,P,P]	C—>M—>P
Posterior likelihood of trial 2	0.51	0.21	0.13

Figure 5.3.2: Results of two-stage inference for each of three G-state models. Each column reports the parameter (error rate for the fixed- and hierarchical-goal models and, and fickleness for the mind-change model) inferred from a single observation of the habituation trial (figure 4.3.1a), a MaP estimate of the actor's goal state for that trial, and the posterior likelihood of the test trial (figure 4.3.1b). Under the fixed-goal model, the observer infers that the actor targets the pie, but has a very high error rate, which reduces the posterior likelihood of the direct path shown in the test trial. Under the mind-change model, the observer infers that the actor made two mind-changes, and therefore has a high fickleness value. This results in lower posterior likelihood of the actor not changing their mind in the test trial. Under the hierarchical goal model, the actor has a low error rate and a 3 step goal sequence. However, the hierarchical model results in a less accurate estimate of the actor's values (since the set of possible goals is much larger), which results in a lower posterior likelihood of the single-step sequence in the test trial

results in several different possible explanations, one for each configuration of goal states compatible with the observation. Furthermore, a larger number of mind changes results in both a higher posterior estimate of ν (the "fickleness" parameter), and a less accurate posterior estimate of the preference parameter (since less preferred outcomes are more likely when the goal state is resampled multiple times). These two factors result in less predictive power and more non-deterministic explanations, and lower posterior likelihood of both the initial and test trials.

The only G-state model which supports a deterministic explanation of the initial trial is a hierarchical model. Under this model, the actor's goal in the initial trial consisted of

three sub-goals, corresponding to picking up the cookie, the muffin, and the pie, in that order, and the posterior probability of this configuration given the data is near 1. However, the space of possible goals under this model is significantly larger than the goal-space under the previous two models, so a single observation provides relatively little information about the actor's preference over goal sub-sequences. As a result, this explanation does not generalize well to the test stimulus, which has relatively low posterior likelihood under the inferred model

It is clear that the two trials in figure 5.3.1 are difficult to coherently explain using G-states alone. In particular, G-states capture information about predicted or counterfactually possible *future* system-states, but do not directly track how the actor's cumulative history of exposure to the relevant environment(s) constrain *current* behavior. We refer to hidden states which track this latter form of information as "B-states," and shall demonstrate that B-states share important structural, representational, and causal features with our commonsense notions of belief, awareness, perception, etc.[59] To this end, we must characterize B-states in the context of our modeling framework.

Like G-states, B-states can take several forms, which we describe in (rough) order from leaner representations to richer. A "lean" B-state consists of a pointer to those features in the current environment of which the actor is currently "aware." This awareness is updated in each step by adding a pointer to any new feature for which the actor currently has perceptual access. This allows the observer to track which features the actor is or is not aware of at time t, but does not require any structured representation of these features (simply a reference or pointer). A "rich" B-state, on the

[59]We shall use the term "beliefs" to refer to this broad folk-psychological category, though for this section we shall focus on perceptual B-states specifically

other hand, is a representation of (some feature of) the actor's epistemic state. The richness of a B-state is, of course, a graded notion, corresponding to the level of information and structural complexity captured by the representation. Importantly, these representations may be underspecified, in that they specify a representation for some part of the environment state, and leave the remaining part unspecified (indicating that the actor is uncertain about the missing part). This uncertainty can also be captured as a probability distribution over possible configurations of the missing system-state information. Examples of lean and rich B-states for the actor in trial 5.3.1(a) are shown in figure 5.3.3

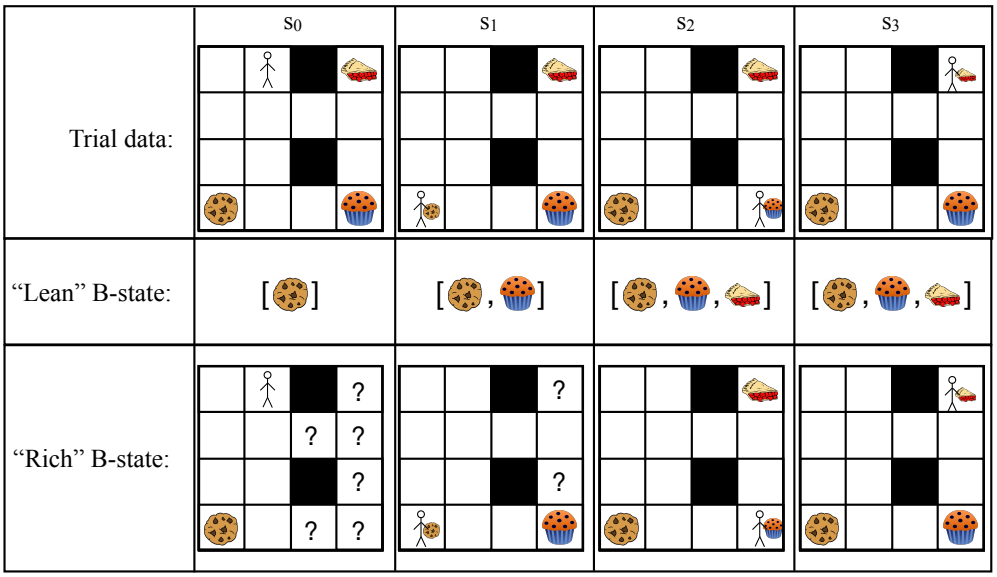


Figure 5.3.3: Example of lean and rich posited B-states for each system-state in trial 4.3.1(a). Panel a) depicts the trial data. Panel b) depicts a lean B-state, which constitutes a list of pointers to those features of which the actor is "aware" in each step. Panel c) depicts a rich B-state, which constitutes an underspecified counterfactual representation of the environment state

Fundamentally, any B-state requires two components:

1. Some form of cumulative information state. We use b_t to denote this state, and the cumulative requirement entails that (some part of) the B-state must be dynamic throughout each episode, i.e. the current b_t must depend on past b_{t-1}'s.

2. A *perceptual access filter* $v(x_t, w_t)$, where we divide the system-state $s_t = (w_t, x_t)$, into a *world state* w_t and *actor state* x_t. Thus, the actor's current bodily state (e.g. where the actor stands in the environment, which direction the actor is facing, etc.) relative to the physical world state (e.g. whether there are visual obstructions in the actor's line of sight) determines what information is added to the actor's B-state in each step. We use θ_b to denote the parameters involved in computing the actor's visual access (e.g. visual range, perceptual accuracy, etc.).

The modular nature of this modeling framework entails that we can construct B-state models by "grafting" B-states onto existing G-state models. However, this does not mean that adding any B-state model to a G-state model will produce a new model with significantly improved predictive or explanatory power. To illustrate this, consider the three actor models shown in figure 5.3.4, which we construct by grafting a B-state onto three different G-state models that we have used in previous examples.

In theory, each of these models should have greater explanatory power than their corresponding G-state-only equivalent. However, only the models in b) and c) provide more coherent explanations of the trials in 5.3.1. To see why, note that, under the fixed-goal model, the actor's goal is sampled at the start of the trial, and remains fixed throughout. Under the B-state-augmented version, this initial goal is dependent on the actor's initial B-state, rather than the initial system-state. However, unless the actor's visual access allows them to see through walls, the actor is not aware of the pie until the

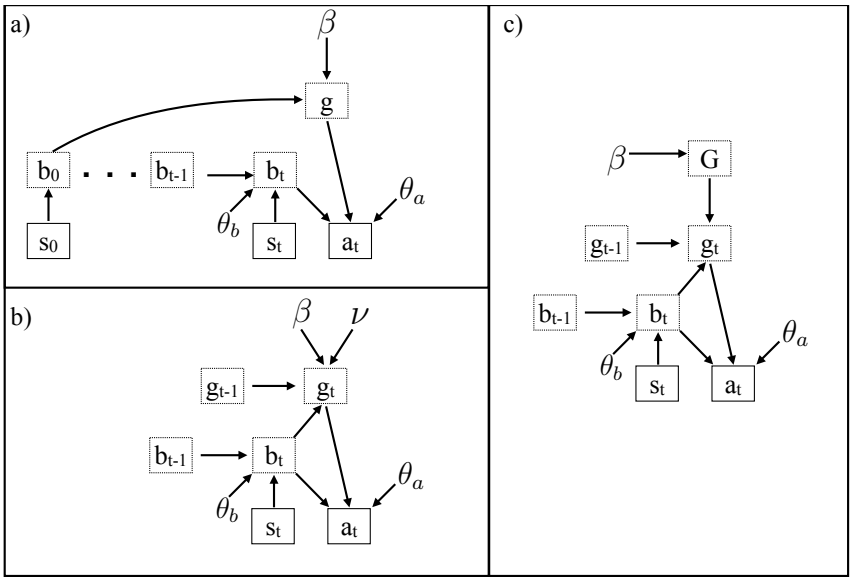


Figure 5.3.4: B-state models adapted from three G-state models which we've previously used in examples. Panel a) depicts an *awareness-constrained* (AC) error-prone fixed-goal model. θ_b encodes the parameters for the actor's B-state update function, and θ_a encodes action parameters (e.g. error-rate, preference for direct paths) Panel b) depicts an AC mind-change model, and ν encodes the probability that the actor will resample their goal upon encountering new information. Panel c) depicts an AC hierarchical goal model

third step. This means that the B-state augmented fixed-goal model provides no coherent explanation of the initial trial.[60] Thus, adding a B-state to the fixed goal model only increases the observer's capacity to explain intermediate behavior, under the assumption that their goal is fixed throughout the trial.

[60]Though this does not mean it *can't* explain the trial, only that the explanation is convoluted and unlikely: e.g. the actor wanted the cookie, but made a long sequence of mistakes and accidentally ended up with the pie, and then the trial ended before they could go back to the cookie

5.3.2 Inferring and reasoning with B-state models

There are two primary ways that B-states can influence and constrain behavior. The first is directly, by constraining the actor's set of possible moves at a given step to the set of *known* moves. For example, there may be two possible paths from the actor's current location to a target state, but a rational actor will still choose the longer path if they lack awareness of the shorter path. The second is indirectly, by influencing the actor's G-state. In particular, every time the actor's B-state is non-trivially updated, that constitutes new information about the actor's environment which may be a relevant input to the actor's G-state update function. For example, if the B-state update includes a new object of which the actor was previously unaware, this presents a new option that the actor may choose from when determining a target outcome, and can therefore resample their goal from the new set of possible options (including choosing an option of which they were previously aware, but had not previously chosen).

The introduction of B-states also introduces the possibility of "epistemic" goals. In particular, an AC actor may not have access to the full set of possible outcomes at the start of a trial. In this case, the actor faces an exploit/explore dilemma: do they take a shortest path to an outcome to which they already have epistemic access, or do they continue exploring the environment in search more desirable outcomes? There are several ways to encode this sort of trade off in an actor model. The most straightforward is to augment the domain of the actor's G-state to include an epistemic *search* goal, which does not point to or represent any specific outcome. If the actor's current goal is *search*, their action function $af(s, search)$ executes an "explore" procedure distinct from the usual shortest-path-to-target procedure executed during rational goal-acquisition.

The explore procedure itself may be structured and strategic, designed to maximize relevant new information in as few steps as possible, or largely random, and the observer can use Level 2 inference to learn about the actor's search behavior specifically.

The G-state update function for such a model must determine when the actor should continue searching, and when they should seek a known target outcome. There are many ways to define and parameterize such a function. A simple method involves a "holdout parameter" ν, which captures the actor's willingness to "hold out" in search of a better option. Given the set $outcomes(B)$ of known possible outcomes in the actor's current B-state, the probability that the actor will continue searching in hope of a better outcome is determined by $\nu * valR(outcomes(B), \beta)$, where $valR(outcomes(B), \beta)$ measures the value potential of other outcomes which may be possible, but which are not currently known to be possible. Intuitively, if the actor's most preferred outcomes are already known, there is a high probability that the actor will stop searching and pursue one of those outcomes. If there are very high value outcomes which are not known to be possible,[61] the actor will be more likely to continue searching and "hold out" for a better outcome. The holdout parameter ν scales the actor's willingness to continue searching: if $\nu = 0$, the actor is completely unwilling to search, and will always pursue the first possible outcome they discover with non-zero value. If $\nu = 1$, the actor's willingness to continue searching for other options is exactly determined by the relative value of those options. We can interpret an actor with $\nu > 1$ as placing intrinsic value on exploration; i.e. an actor with very high ν value may continue searching even if they have already discovered all of their highest-value outcomes.[62] An example of such an actor model is

[61] And are not known to be impossible

[62] In more general cases, where we represent the actor's G-state as a value function

shown in figure 5.3.5.

Given an actor model of this form, the observer can infer an actor's willingness to hold out in search of a better option via Level 2 inference. If the observer already has a good estimate of the actor's preference parameter β, the observer can track whether the actor tends to move directly toward a high-value target as soon as they become aware of that target (indicating a low ν value), or whether the actor tends to continue exploring after discovering a high-value target (indicating a high ν value). If the observer does not already know the actor's preferences, they can use the same Level 2 program to infer a joint distribution over both. These estimates will generally be correlated: if the β estimate assigns higher value to outcomes which the actor does not immediately lock onto (i.e. continues searching after discovering that outcome), this should correlate with higher estimates of ν, and visa versa. That is, an actor who continues searching even

---

over states, we can encode epistemic goals by augmenting the G-state value function with the actor's intrinsic epistemic values. That is, we can design a single value function v, parameterized by both the actor's preferences over outcomes β and an epistemic value parameter ν. The value $v(s; \beta, \nu)$ that this function assigns to a system-state s depends on both the reward value of that state as an outcome (determined by β), as well as the epistemic value of the new information that would be reaped in that state (determined by ν). By encoding intrinsic epistemic values into the actor's general value function, we can avoid having to define two distinct "modes" of behavior (search and acquire), and instead solve the actor's explore-exploit trade off by maximizing the total discounted utility of this single value function. However, as we will mostly confine our examples to those involving sparse value functions, we will not fully explore this alternative in the scope of this dissertation.

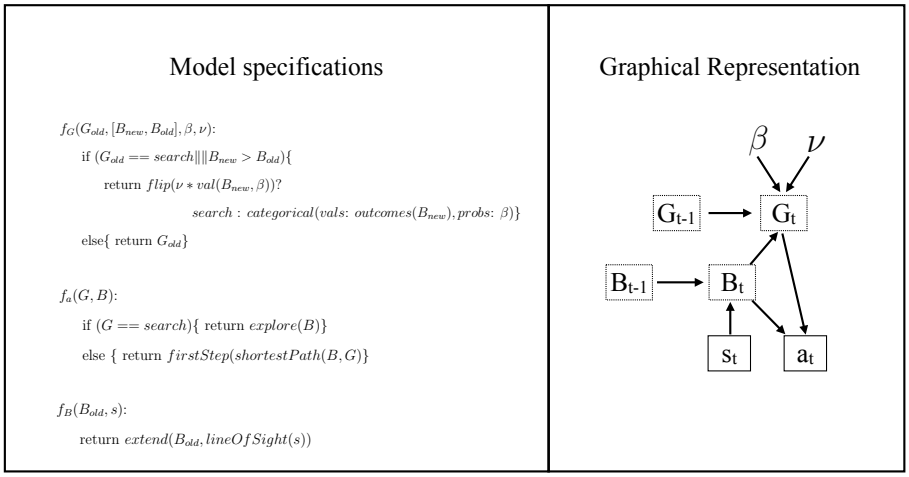


Figure 5.3.5: Example of an AC simple goal model. The G-state update function first determines the probability that the actor will continue searching. This is determined by the value of currently known possible outcomes, relative to the value of possible remaining outcomes, scaled by a "holdout" parameter ν. If the actor's current goal is set to *search*, they execute a distinct *explore* procedure, depending on their current access B to the environment. Otherwise, they execute a shortest-path search to a target outcome

after discovering a more highly valued outcome must be more willing to continue searching for other options.

This characterizes how B-states can (directly or indirectly) constrain the actor's behavior. However, a B-state model also needs to specify how the actor's B-states are updated. This depends on a "perceptual access" filter $v(x_t, w_t)$, where we divide the system state $s_t = (w_t, x_t)$ into the *world state* (e.g. layout of the current environment) and the *actor state* (e.g. actor's current location and the direction they are facing). For example, in the actor model shown in figure 5.3.5, this corresponds to the "line of sight" function, which scans the actor's line of sight, determined by their current position and direction, until it reaches a wall or the edge of the grid. The B-state update function then adds any newly discovered features to the actor's previous B-state. The parameters

in the perceptual access filter encode information about the actor's perceptual constraints: for visual access, this includes information like the actor's visual range (e.g. an actor with poorer vision might not be able to read a clock that is far away), visual accuracy (e.g. an actor who is known to be red-green colorblind may miss certain color-coded information), and other relevant constraints. Given a particular parametric form for the visual access filter, these parameters can be learned via Level 2 inference. More generally, it may be possible to adopt a generic, high-dimensional parametric form (e.g. neural network) to learn $v(x_t, w_t)$ in a model-free way.

Thus far, we have described how an observer can learn the parameter values for a particular B-state model applied to a particular actor. A more fundamental question, however, is how the observer infers that the actor is awareness constrained (AC) in the first place. Intuitively, an AC actor produces distinctive behavioral patterns compared to a non-AC (omniscient, or Omni) actor. Whereas an Omni actor should consistently take a shortest path to a particular outcome, an AC actor will likely take a significantly longer path to their final outcome, reflecting either their search for a path to the outcome, or their search for better alternatives (or both). To illustrate how the observer can leverage this to recognize that an actor is AC, we perform a series of simulations involving a simple framework theory that contains two possible model structures: an omni-rational error-prone goal seeker (who knows all of the possible outcomes and all possible paths to those outcomes at the start of each trial), and an AC-rational error-prone goal seeker, with a lean (list of features) B-state and holdout parameter. To illustrate the stark distinction between AC behavior and Omni behavior, we generate a data set using only 5 trials, generated by a single actor (who is, in fact, AC). The results of this inference are shown in figure 4.3.6.

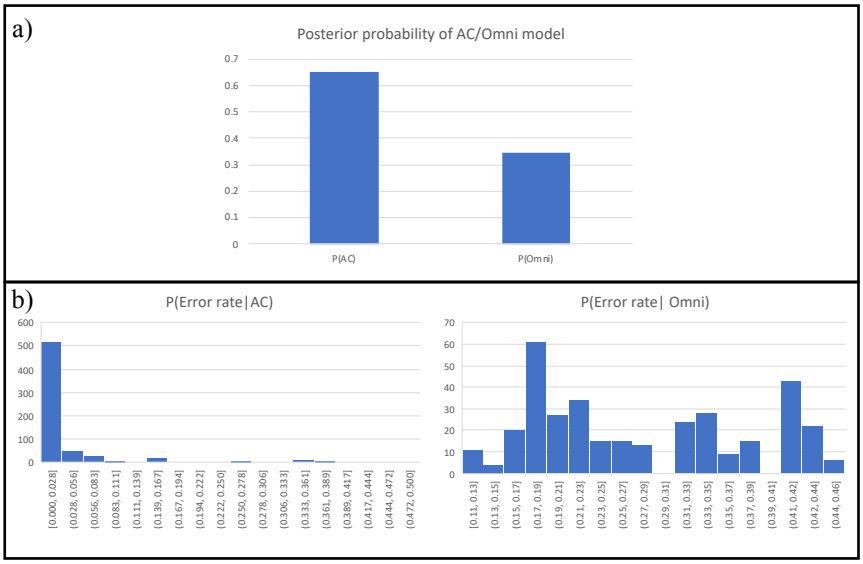


Figure 5.3.6: Results of a Level 2 experiment to infer whether the actor is AC or Omni. Panel a) depicts the inferred posterior probability that the actor is AC or Omni, which is significantly higher for the AC model. Panel b) depicts the posterior distributions over error rate, conditioned on each model type. The error distribution for the AC model is skewed significantly toward lower values, while the Omni distribution is skewed towards higher values. This reflects that the only way to explain this data using an Omni actor model is to attribute an extremely high error rate to the actor

In addition to showing the posterior probability of each actor type (AC or Omni), we show the posterior distribution over error rates associated with each model type. This reflects that, although the Omni explanation is significantly less likely, it is still possible to explain the data using an Omni model. This explanation, however, would require a significantly higher error rate, as the depicted behavior often involves non-direct paths to the final outcome. Thus, the only way to explain this data using an Omni model is to assume that the actor is *extremely* clumsy. This means that the inferred Omni model provides less predictive power and less deterministic explanations that the AC model, so the observer strongly prefers the AC explanation over the Omni explanation.

5.3.3 Developing and revising a B-state theory

As we described at the beginning of 5.3.1, the developmental data make it clear that our ability to infer and reason about epistemic states does not emerge all at once. One possible explanation of this gradual development is that younger children possess a non-representational understanding of epistemic states, which tracks something more primitive that full-fledged belief states. Intuitively, the younger child's theory tracks an actor's epistemic access to/awareness of an environment, while the older child's theory tracks how the actor represents the environment.[63] Thus, under this theory, the older child's performance in a false belief is possibly a result of additional representational resources not available to the younger child. This is further supported by the observation that younger and older children can track whether an actor is aware of an object, but only older children can track *how the object appears* to the agent (Masangkay et al 1974).

To explore this possibility, we can model the two theories using lean and rich B-states. As we described in the previous section, a lean B-state tracks the actor's access to or awareness of an environment: a simple way to encode this is as an $n \times 1$ binary vector b, where b_i denotes whether or not the actor is aware of feature i.[64] A lean B-state is updated by changing b_i to 1 whenever feature i is detected by the actor (which is determined by their visual access function). A rich B-state tracks how the actor *represents* the current state. This can be encoded as an underspecified representation of the system, with "missing" entries to represent the actor's uncertainty (e.g. the contents of a closed, opaque box). It may also specific a probability distribution over the

[63]Or, more efficiently, the *differences between* the observer's own representation of the environment and the actor's representation

[64]Or as a list of pointers to the known features, as shown in figure 5.3.3

underspecified portion, capture the actor's "degree of belief" for each possibility. The actor's visual access function determines the "portion" of the B-state that is updated in each step.

The process through which a B-state constrains the actor's behavior differs for lean and rich B-states. For lean models, the B-state imposes a filter over the set of possible actions. In particular, if a computation in the actor's action-selection function requires the value of feature i, and the actor is not aware of feature i, then the actor may not take any action which requires this computation.[65] For example, the goal-seeker models we described in section 5.2 make use of a shortest-path program $shortestPath(s, g)$, which computes the shortest path from state s to a state with feature g. If the actor is unaware of feature g at time t, the actor may not compute that path, and may only take an optimal action by mistake (or as a search behavior). In a rich model, the B-state can be "plugged into" the actor's action selection function in lieu of the system state. That is, the actor behaves as though the represented system state *is* the real system state, and resolves missing values in that system state using the associated probability distribution.

Putting this all together, we can test the performance of lean and rich models in an MDP version of the false belief task. Figure 5.3.7 shows the inferred B-states and predicted final actions under the lean and rich B-state models. Under the lean model, the actor's B-state tracks which features (muffin, box, and basket) the actor is aware of at time t. When computing the final action, the B-state indicates that the actor is still "aware" of the muffin, and can therefore compute the shortest path to that target, thus

[65]Note that the actor may still take that action if they can compute it another way without relying on the missing value. E.g. the actor could still take that move by accident, or as a result of exploration behavior

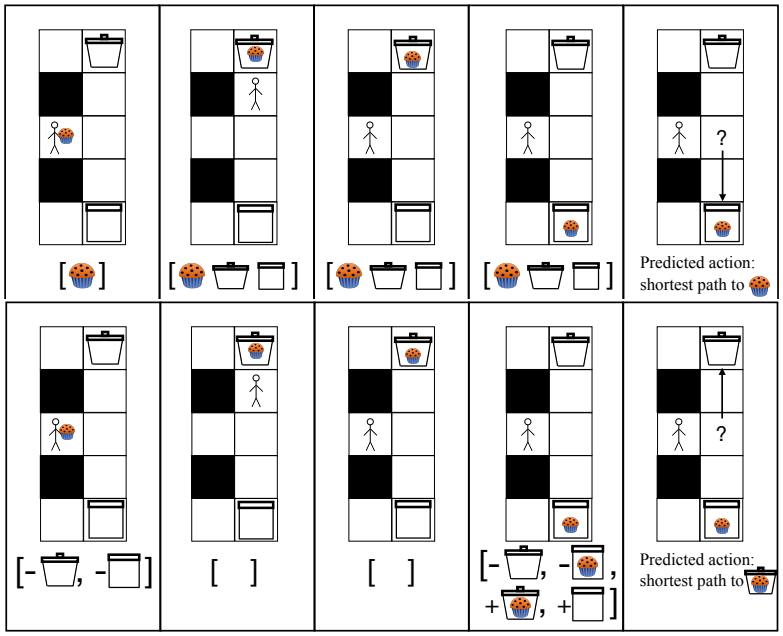


Figure 5.3.7: Explanation and final prediction for MDP false-belief test. Top panel depicts inferred B-states and predicted final action under a lean model, which tracks the actor's awareness in a non-representational fashion. Bottom panel depicts inferred B-states and predicted final action under a rich model, which tracks how the actor represents the system. We represent the rich B-state as a difference between the observer's own belief (an accurate representation of the grid) and the actor's belief

resulting in the "incorrect" prediction, which corresponds to the performance of younger children. The rich model, however, tracks how the actor represents the environment. We use "difference" notation in figure 5.3.7, by listing only those features which differ between the actor's representation and true system state. When computing the final action, the actor targets the container which still (according to their representation) contains the muffin (the basket). Thus, the rich model results in the "correct" prediction, corresponding to the performance of older children.

5.4 Discussion

In the first section of this chapter, we defined a set of social inference problems (in terms of the data and task requirements), an inference agent with certain computational constraints, resource constraints, and biases. We then showed, through analysis and simulation, that:

1. Given the observer's task demands, constraints, and biases, purely behaviorist actor models require significantly more parameters than cognitive actor models, and provide less deterministic explanations than cognitive actor models. Thus, a psychological framework theory is a more rational solution than a purely behaviorist framework theory.

2. In order to explain human behavioral data, a psychological framework theory must include a type of hidden state (a G-state), which shares important functional and representational features with our commonsense notions of desires/values.

3. G-states alone do not adequately explain certain kinds of behavioral data, and the rational solution to this problem is to include another kind of hidden state (a B-state) which shares important functional and representational features with our commonsense notions of beliefs/awareness.

Thus, we have presented an argument to the effect that Belief-Desire-Action folk psychologies are, in fact, rational solutions to social inference problems, which satisfies the first goal we raised in Chapter 1. Furthermore, we were able to formalize the many different conceptions and representations that fall under our commonsense concepts of "beliefs" and "desires," and demonstrate why (and under what circumstances) the

rational solutions to social inference problems involve hidden states with these features. In conjunction with our Level 3 inference framework, this illustrates the plausibility of the notion that we develop BDA folk psychologies *because* they are rational, through an (approximately) rational inference process. This addresses the second goal we raised in Chapter 1.

To address the third goal we raised in Chapter 1, we drew on several well-known examples from developmental psychology to motivate how certain transitions apparent in our ToM development may reflect some form of rational theory revision. In 5.2, we showed how a transition in infants' expectations about goal-directed behavior can be replicated by simulating an inductive transition from a non-mentalistic (teleological) theory of goal-directed action to a mentalistic theory. We also demonstrated how an increased concentration of "unfulfilled goal" episodes in the observer's data could induce such an inductive transition. Finally, in section 5.3, we addressed the hypothesis that the differences between younger and older childrens' performance in belief-inference tasks can be explained by the "richness" or "lean-ness" of the attributed epistemic states. In particular, we presented a "lean" form of belief state which captures certain information about the actor's epistemic access, but results in incorrect behavior predictions in a false belief test. This suggests that our ability to reason about false beliefs may depend on certain representational resources not available to younger children.

6 Conclusions and future work

6.1 Summary

As we discussed in chapters 1 and 2, Theory of Mind is deeply entrenched in human language and cognition, and understanding how ToM develops is important for understanding human cognitive development in general. There is a wealth of data from developmental psychology that tracks how ToM develops throughout childhood, but the nature of these data make it difficult to explain, from a theoretical standpoint, how this development occurs. On the one hand, the data show a clear and cross-culturally consistent ordering in the mental reasoning skills that children acquire. On the other hand, there is a relatively high degree of within-cultural *and* cross-cultural variability in the ages at which different children acquire these skills. Furthermore, cross-cultural data reveal a few noticeable deviations from the most typical developmental trajectories, which seem to be tightly correlated with specific cultural or linguistic inputs. Thus, the data reveal that our ToM development is clearly sensitive in some ways to our social data inputs, but at the same time, that development very consistently results in the same Belief-Desire-Action heuristic structure (though not necessarily in the same order). This makes ToM development challenging to explain from a strictly nativist or empiricist perspective.

To address these challenges, we proposed a computationally grounded *rational constructivist* account, according to which ToM is essentially pieced together from primitive conceptual components through a process that resembles approximately rational hypothesis revision. We laid out a formal framework for developing such an account in Chapter 3, which draws on the formal machinery of Bayesian Theory of Mind,

Hierarchical Bayesian Models, and functional probabilistic programs. The models in this framework represent a single observer agent learning and reasoning about one or more actors. Importantly, our framework distinguishes between three hierarchical levels of reasoning. Level 1 problems involve explaining and reasoning about a particular instance of a particular actor's behavior. Level 2 problems involve learning about a particular actor's general values, preferences, cognitive constraints, and behavior patterns, and this knowledge is used as an input to Level 1 inference. Level 3 problems involve learning about agent behavior in general from observations of multiple different agents. The "framework theory" that results from Level 3 inference serves as a template for learning a Level 2 "actor model," and the Level 1 "trial explanation" inherits its structure from the Level 2 model. The purpose of this modeling framework is two-fold. On the empirical side, the models are used as a framework for interpreting data collected in a cognitive-behavioral experiment, and assessing what we can infer about a subject's cognitive representations from their behavioral responses. On the theoretical side, the models are used in simulation to demonstrate how and under what conditions certain patterns of development are possible/plausible/likely.

In Chapter 4, we present a methodological framework for empirical applications of these models. We focus in particular on infant studies for two reasons. First, the very early stages of development are especially crucial for understanding human cognition, and there is a wealth of experiments on ToM in infants. Second, infant cognitive studies face some severe methodological challenges as a result of the extreme sparsity of data, and our framework is highly useful for addressing these challenges. In particular, because young infants cannot report or describe their own cognitive representations, we must rely on a "linking hypothesis" which relates the infants' behavioral response (typically visual

fixation) to a cognitive process or representation. To this end, we demonstrate how our framework can be used to simulate the standard two-stage visual habituation paradigm under different cognitive representations, and compare the test-phase performance against infants' actual responses to the same stimuli. This provides theoretical justification for a claim of the form: "A subject with theory (cognitive representation) T, if habituated to stimulus s, will show response a to test stimulus t." We can then invert this reasoning to conclude that a subject who does *not* show response a to test stimulus t when habituated to s does *not* interpret the class of stimuli using theory T. We illustrate how this principle can provide a more refined interpretation of infant visual fixation data by replicating, in simulation, a seminal study into infants' understanding of goal-directed behavior (Woodward 1998).

In Chapter 5, we explore the theoretical applications of our framework by modeling a hypothetical developing human child as a kind of social inference agent. This agent is exposed to certain kinds of social data (using the MDP representation system), tasked with certain inference problems (predicting and explaining actor's behavior), under certain computational and resource constraints, with certain inductive biases (for "simple" and "deterministic" explanations). We then show, analytically and through simulation, that given these problems and constraints, solutions (i.e. framework theories) that share certain features with our Belief-Desire-Action (BDA) folk psychology are more rational than those without. In particular, we show how

- "behaviorist" theories (i.e. those that do not posit hidden states) are less rational (per our constraints) than "cognitive" theories
- "cognitive theories" require certain kinds of hidden states (G-states) in order to

adequately explain human behavioral data, and these states share important representational and relational features with our commonsense notions of "goals"

- G-states alone are not sufficient to explain certain kinds of behavioral data, and the rational revision to this data involves adding a distinct type of hidden state (B-states), which share important representational and relational features with our common-sense notions of "beliefs."

In addition to showing that BDA-like solutions are more rational than non-BDA solutions, we drew on our Level 3 framework to demonstrate how a domain general inference mechanism (simulation and sampling of hierarchical generative models) can leverage this notion of rationality (encoded by certain biases in the observer's prior distribution) to "learn" a BDA-like theory of mind from cross-agent behavioral data. Finally, we use our Level 2 and Level 3 framework to a) demonstrate that certain transitions apparent in our ToM development may reflect different stages of underlying framework theory, and b) motivate how these transitions may in fact reflect rational theory revision in response to new data or constraints.

6.2 Future work

6.2.1 Improving the framework

While we have presented the foundations for a constructivist account of ToM development, there are several aspects which need to be further developed in the future.

Fleshing out and unifying the observer's constraints In chapter 5, we defined several kinds of constraints on the observer which, in conjunction with specific task

demands, determine a normative notion of "rationality" over the set of solutions (framework theories/actor models/explanations). The first kind were computational constraints, i.e. a set of primitive representations (e.g. system-state representations, value functions, etc.) and a set of manipulations (recursive composition, marginalization, conditioning) that the observer can perform over these representations. The second kind were "simplicity" constraints, which we can think of more generally as a type of resource constraint. The third kind were intrinsic biases, e.g. the observer's bias for deterministic explanations.

However, rather than fully specifying and unifying each of these dimensions into a single measure, we only considered trade-offs and interactions along individual dimensions. For example, we observed that behaviorist framework theories are both less simple than cognitive theories and provide less deterministic explanations, but we did not consider general interactions between simplicity and the capacity for deterministic explanation, nor did we examine how simplicity and determinism may complement each other/trade off in a way that can be leveraged to optimize task performance. In order to perform this more rigorous and general analysis, we will need to both a) flesh out the observer's cost and simplicity constraints in greater detail, and b) unify the observer's different kinds of constraints into (ideally) a single measure that can be used in rational optimization problems. The first part will involve a more detailed decomposition of the observer's total "resource" use: this includes the memory/information costs of retaining a theory, the memory/information costs of retaining the actor-specific information used to construct an actor model, the computational cost of inferring and manipulating an explanation during Level 1 inference, etc. Thus, there are clearly many dimensions that would need to be considered and integrated into a single "cost" measure.

One promising framework for deriving and using such a measure is *resource rational analysis* (Lieder & Griffiths 2020). The resource rationality paradigm extends the rationalist approach to cognitive modeling to the algorithmic and implementation levels of analysis. Under the standard computational-level approach, a rationalist model makes minimal assumptions about the observer's cognitive constraints, and is used to interpret human cognitive behavior as an approximately rational response to some environmental problem. The *resource rationality* approach seeks to explain human cognitive behavior as a rational *allocation of limited computational resources* in response to an environmental problem. The core components of an observer model in this framework are

1. An environmental problem (i.e. set of environments, observations in environments, possible responses to each environment, value function over environment/response pairs)
2. A set of possible programs mapping observations to actions
3. A measure of the "cost" associated with storing and using each program
4. A measure of the "value" derived from relying on each program in each environment

Given these components, we can define the "rational program" for an environmental problem as (roughly) the one which maximizes the observer's lifetime performance in the problem for the lowest possible cost. If we can express our observer's problems and constraints in this form, we can leverage the resource rationality approach to derive a more rigorous, unified rationality principle for ToM inference. In order to do this, we

must translate each of the four components listed above into the context of our framework:

1. An environmental problem corresponds to the set of social inference problems faced by the observer (data and tasks)

2. The set of possible programs is determined by the observer's computational resources, in this case a set of stochastic and representational primitive functions, which can be composed and manipulated using stochastic recursion, marginalization, and conditioning.

3. The observer's simplicity constraints both translate into types of cost constraints on programs. We previously suggested that the number of trainable parameters in a model determines the cost of storing actor-specific information, while the number of variables in a model determines the cost of storing the general model template. This is an oversimplification, but serves as a useful initial direction for deriving a proper cost function. We will additionally need to account for the domain of each parameter and variable, and the expected number of parameters that are actually needed for a given inference (e.g. a model may encode lots of preference and value parameters, but only ever need a handful of those parameters at any given time).

4. The "value" derived from relying on each program must reflect both the value generated by a particular response (e.g. the observer's reward for a correct prediction/penalty for an incorrect prediction), and the observer's own intrinsic values or biases, in this case a bias for determinism[66]

[66]Though it is possible that this bias could be accounted for purely in terms of the

This outlines a path towards a more rigorous and unified treatment of the observer's constraints and biases.

Learning more complex theories The examples of Level 3 inference that we presented in this dissertation dealt with fairly constrained problems consisting of either a) learning a hyper-parameter for a theory or b) choosing between a handful of fixed theories. This is useful for illustrating why an observer would prefer one theory over another, but it does not fully address *where* the theories came from in the first place. That is, how does the observer construct these candidate theories out of more basic primitives? In Chapter 3.5, we suggested that Probabilistic Generative Grammars and/or non-parametric priors can be used to "build" a theory of arbitrary complexity out of component functions. If used for a Level 3 simulation with sufficient cross-actor data, a PGG would allow the observer to perform such a construction, which would provide strong justification that the candidate theories themselves could be learned from data using the same basic mechanisms. We outline a simple PGG for rational actor models in appendix B4, but there is a challenge that must be resolved in order to generalize this further. This is the challenge of composing and modifying the component programs that relate entities in a theory, and doing these compositions in a way that guarantees the resulting program will run from end-to-end.

Suppose, for example, that our observer starts with a fixed-goal error-prone theory. This theory uses a component program which computes the shortest path to the goal, and (noisily) pursues that path. If the observer were to modify this theory by adding,

observer's resource constraints, i.e. that the observer has a bias for determinism *because* programs which entail deterministic explanations are less costly to store and use

say, a REP variable corresponding to a belief state as an input to actions, the observer must modify their action program to accommodate this new input. Thus, our PGG or non-parameteric prior has to specify how each transformation to the model structure modifies the affected component programs (i.e. any variable that gained or lost an input), and it must specify these program transformations in a way that does not "break" the theory (or at least, places extremely low prior probability on "broken" programs). In our initial example, we attempted to do this by making an additional assumption about the observer's computational primitives: in particular, we added a primitive "expected utility" function[67] that the observer can use to compute the expected future value of a particular action (weighted by the observer's own estimates of the actor's future actions). This provides a sufficiently general program form that we can define a grammar which modifies component programs as well as the corresponding model structures.

A more general approach, which would not require this fairly strong assumption, would be to use the probabilistic grammar to generate the model structure alone, then use an all-purpose function estimator (e.g. a neural network) to infer the programs that link each component. Indeed, neural network techniques have been applied to social cognition and Theory of Mind (e.g. Rabinowitz et al 2018), but these applications are generally intended to show that neural networks can be trained to perform ToM tasks, rather than showing how human beings do, in fact, learn to perform ToM tasks. While deep neural networks are extremely flexible and can provide predictive accuracy in practically any task, they are notoriously opaque[68] and generally don't reveal what the

[67]In the simple grid-world examples, this reduces to the path-minimization program

[68]While there is research into "opening the black box" of neural networks, many of these post-hoc analysis techniques are specific to certain tasks and still require additional

network has "learned" or what the network can tell us about human cognition. Due to the generality of PPLs, however, we can define a PGG in which the component programs mediating variable interactions all use a default function estimator form, without having to change our framework definitions. Under this hybrid approach, Bayesian inference is used to learn the "high level architecture" of the theory (i.e. the ontology of mental states and dependency relations), and standard function-estimation methods can be used to learn the low-level programs that mediate these states. This would allow us to define a more general PGG without having to attribute strong prior knowledge to the observer (e.g. the capacity to do discounted expected future utility computations). Intuitively, this approach may allow us to simulate very complex theory-learning problems more efficiently by marginalizing out the lower-level parameter and program information.

6.2.2 Empirical assessment of theoretical developmental hypotheses

In chapter 4, we presented a methodological framework for interpreting the data generated by cognitive behavioral experiments using our computational models. Importantly, the framework we presented was designed to test hypotheses about a subject's "intuitive theory" or representation of a class of stimuli by simulating Level 2 inference for a given intuitive theory. In particular, our framework allows us to simulate how a rational observer with intuitive theory T (about a given domain of stimuli) would respond to a test stimulus in that domain. Thus, this framework is useful for evaluating a subject's intuitive theory or representation of a kind of stimulus.

In chapter 5, however, we hinted at a methodology for testing a different kind of hypothesis: in particular, the hypothesis that an observed transition in children's social

human interpretation to make sense of in a general way

reasoning capacities reflects an inductive revision in the child's intuitive theory of mind. At a high level, this methodology involves the following steps:

1. Identify a class $\mathbb{S}$ of ToM or social reasoning tasks in which children belonging to age group X respond in a systematically different way than children belonging to age group Y

2. Identify sets $\mathbb{T}_1$ and $\mathbb{T}_2$ of framework theories such that a rational observer with a theory in $\mathbb{T}_1$ responds to a task in $\mathbb{S}$ like an X-year old, while a rational observer with a theory in $\mathbb{T}_2$ responds like a Y-year old.

3. Use Level 3 inference to determine whether, and under what conditions, a rational observer would transition from a theory in $\mathbb{T}_1$ to $\mathbb{T}_2$.

The third step can be applied in two directions: we can use theoretical insights to identify, in simulation, factors which seem to induce an appropriate inductive revision, then examine whether children encounter similar factors between the ages of X and Y. In the other direction, we can draw on observations (or suspicions) about changes that children seem to face between the ages of X and Y, then introduce these changes to our observer to determine when it induces an appropriate revision. If we have reason to believe, for example, that 5-year olds are able to pass false belief tests because they have additional representational resources that 4-year olds have not yet developed, we can test this in simulation by defining an observer with limited representational resources, and observing how they respond when we exogeneously augment those resources.

The ultimate goal in this approach would be to identify a single inductive trajectory that explains a large range of childhood ToM development (i.e. a long sequence of

transitions that we observe throughout childhood), and replicate this trajectory in simulation. Doing so would provide theoretical insights into the intuitive theories and cognitive representations that children draw on in each stage of development, and we could empirically assess each individual claim (i.e. that children with response pattern a have theory T) using the methodology from chapter 4. Fortunately, there is a significant literature on scaling ToM tasks, and several groups of authors have developed sequences of mental inference problems which seem to capture cumulative stages of children's ToM development (e.g. Wellman & Liu 2004). An obvious and fruitful direction for future work is to identify inductive trajectories of framework theories which replicate these scaled stages of development.

References

[1] Abbeel, P., & Ng, A. Y. (2004, July). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning* (p. 1).

[2] Ashmead, D. H., & Davis, D. L. (1996). Measuring habituation in infants: An approach using regression analysis. *Child Development*, 67(6), 2677-2690.

[3] Aslin, R. N. (2007). What's in a look?. *Developmental science*, 10(1), 48-53.

[4] Aslin, R. N., & Fiser, J. (2005). Methodological challenges for understanding cognitive development in infants. *Trends in cognitive sciences*, 9(3), 92-98.

[5] Baillargeon, R. (1986). Representing the existence and the location of hidden objects: Object permanence in 6-and 8-month-old infants. *Cognition*, 23(1), 21-41.

[6] Baillargeon, R., Scott, R. M., & He, Z. (2010). False-belief understanding in infants. *Trends in cognitive sciences*, 14(3), 110-118.

[7] Baker, C., Saxe, R., & Tenenbaum, J. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 33, No. 33).

[8] Baker, C. L., Saxe, R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, 113(3), 329-349.

[9] Bashinski, H. S., Werner, J. S., & Rudy, J. W. (1985). Determinants of infant visual fixation: Evidence for a two-process theory. *Journal of Experimental Child Psychology*, 39(3), 580-598.

[10] Behne, T., Carpenter, M., Call, J., & Tomasello, M. (2005). Unwilling versus unable: infants' understanding of intentional action. *Developmental psychology*, 41(2), 328.

[11] Bernstein, A. S. (1979). The orienting response as novelty and significance detector: Reply to O'Gorman. *Psychophysiology*, 16(3), 263-273

[12] Bernstein, A. S. (1981). The orienting response and stimulus significance: Further comments. *Biological Psychology*, 12(2-3), 171-185.

[13] Brandone, A. C., & Wellman, H. M. (2009). You can't always get what you want: Infants understand failed goal-directed actions. *Psychological science*, 20(1), 85-91.

[14] Brooks, R., & Meltzoff, A. N. (2002). The importance of eyes: how infants interpret adult looking behavior. *Developmental psychology*, 38(6), 958.

[15] Brooks, R., & Meltzoff, A. N. (2005). The development of gaze following and its relation to language. *Developmental science*, 8(6), 535-543.

[16] Buchsbaum, D., Bridgers, S., Skolnick Weisberg, D., & Gopnik, A. (2012). The power of possibility: Causal learning, counterfactual reasoning, and pretend play. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1599), 2202-2212.

[17] Butler, S. C., Caron, A. J., & Brooks, R. (2000). Infant understanding of the referential nature of looking. *Journal of Cognition and Development*, 1(4), 359-377.

[18] Chater, N., & Oaksford, M. (2013). Programs as causal models: Speculations on mental programs and mental representation. *Cognitive Science*, 37(6), 1171-1191.

[19] Chomsky, N. (2006). On cognitive structures and their development: A reply to Piaget. *Philosophy of mind: Classical problems/contemporary issues*, 751-755.

[20] Chomsky, N. (1986). Knowledge of language: Its nature, origin, and use. *Greenwood Publishing Group*.

[21] Cohen, L. B., & Oakes, L. M. (1993). How infants perceive a simple causal event. *Developmental Psychology*, 29(3), 421.

[22] Colombo, J., & Mitchell, D. W. (2009). Infant visual habituation. *Neurobiology of learning and memory*, 92(2), 225-234.

[23] Csibra, G., & Gergely, G. (1998). The teleological origins of mentalistic action explanations: A developmental hypothesis. *Developmental Science*, 1(2), 255-259.

[24] Dannemiller, J. L. (1984). Infant habituation criteria: I. A Monte Carlo study of the 50% decrement criterion. *Infant Behavior & Development*.

[25] D'Entremont, B., Hains, S. M., & Muir, D. W. (1997). A demonstration of gaze following in 3-to 6-month-olds. *Infant Behavior and Development*, 20(4), 569-572.

[26] Dunham, P., Dunham, F., & O'Keefe, C. (2000). Two-year-olds' sensitivity to a parent's knowledge state: Mind reading or contextual cues?. *British Journal of Developmental Psychology*, 18(4), 519-532.

[27] Fantz, R. L. (1958). Pattern vision in young infants. *The psychological record.*

[28] Fantz, R. L. (1961). A method for studying depth perception in infants under six months of age. *The Psychological Record*.

[29] Fantz, R. L. (1964). Visual experience in infants: Decreased attention to familiar patterns relative to novel ones. *Science*, 146(3644), 668-670.

[30] Farroni, T., Csibra, G., Simion, F., & Johnson, M. H. (2002). Eye contact detection in humans from birth. *Proceedings of the National academy of sciences*, 99(14), 9602-9605.

[31] Flavell, J. H., Green, F. L., Flavell, E. R., Harris, P. L., & Astington, J. W. (1995). Young children's knowledge about thinking. *Monographs of the society for research in child development*, i-113.

[32] Frank, M. C., Vul, E., & Johnson, S. P. (2009). Development of infants' attention to faces during the first year. *Cognition*, 110(2), 160-170.

[33] Gergely, G., Nádasdy, Z., Csibra, G., & Bíró, S. (1995). Taking the intentional stance at 12 months of age. *Cognition*, 56(2), 165-193.

[34] Gerstenberg, T., Goodman, N., Lagnado, D., & Tenenbaum, J. (2014). From counterfactual simulation to causal judgment. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 36, No. 36).

[35] Gnepp, J. (1983). Children's social sensitivity: Inferring emotions from conflicting cues. *Developmental Psychology*, 19(6), 805.

[36] Goodman, N. D., Baker, C. L., Bonawitz, E. B., Mansinghka, V. K., Gopnik, A., Wellman, H., ... & Tenenbaum, J. B. (2006, July). Intuitive theories of mind: A rational approach to false belief. In *Proceedings of the twenty-eighth annual conference of the cognitive science society* (Vol. 6). Vancouver: Cognitive Science Society.

[37] Goodman, N.D., Tenenbaum, J.B., & the ProbMods Contributors (2016). Probabilistic Models of Cognition (2nd ed.)

[38] Goodman, N. D., Ullman, T. D., & Tenenbaum, J. B. (2011). Learning a theory of causality. *Psychological review*, 118(1), 110.

[39] Gopnik, A., Meltzoff, A. N., & Bryant, P. (1997). Words, thoughts, and theories (Vol. 1). Cambridge, MA: Mit Press.

[40] Gopnik, A., & Slaughter, V. (1991). Young children's understanding of changes in their mental states. Child development, 62(1), 98-110.

[41] Gopnik, A., & Wellman, H. M. (2012). Reconstructing constructivism: Causal models, Bayesian learning mechanisms, and the theory theory. *Psychological bulletin*, 138(6), 1085.

[42] Gopnik, A., & Wellman, H. M. (1992). Why the child's theory of mind really is a theory. *Mind & Language*, 7(1-2), 145-171.

[43] Griffiths, T. L., & Austerweil, J. L. (2009). Analyzing human feature learning as nonparametric Bayesian inference. In *Advances in neural information processing systems* (pp. 97-104).

[44] Groves, P. M., & Thompson, R. F. (1970). Habituation: a dual-process theory. *Psychological review*, 77(5), 419.

[45] Hamlin, K. J., Ullman, T., Tenenbaum, J., Goodman, N., & Baker, C. (2013). The mentalistic basis of core social cognition: Experiments in preverbal infants and a computational model. *Developmental science*, 16(2), 209-226.

[46] 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(2), 172-200.

[47] Horowitz, F. D., Paden, L., Bhana, K., & Self, P. (1972). An infant-control procedure for studying infant visual fixations. *Developmental Psychology*, 7(1), 90.

[48] 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(8), 589-604.

[49] Johnson, M. H., Dziurawiec, S., Ellis, H., & Morton, J. (1991). Newborns' preferential tracking of face-like stimuli and its subsequent decline. *Cognition*, 40(1-2), 1-19.

[50] Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. Behavioral and Brain Sciences, 34(4), 169-188.

[51] Kemp, C., & Xu, F. (2009). An ideal observer model of infant object perception. In *Advances in Neural Information Processing Systems* (pp. 825-832).

[52] Kersten, D., & Yuille, A. (2003). Bayesian models of object perception. *Current opinion in neurobiology*, 13(2), 150-158.

[53] Kidd, C., Piantadosi, S. T., & Aslin, R. N. (2012). The Goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PloS one*, 7(5), e36399.

[54] Kovács, Á. M., Téglás, E., & Endress, A. D. (2010). The social sense: Susceptibility to others' beliefs in human infants and adults. *Science*, 330(6012), 1830-1834.

[55] Kushnir, T., Gopnik, A., Chernyak, N., Seiver, E., & Wellman, H. M. (2015). Developing intuitions about free will between ages four and six. *Cognition*, 138, 79-101.

[56] Kushnir, T., Xu, F., & Wellman, H. M. (2010). Young children use statistical sampling to infer the preferences of other people. *Psychological science*, 21(8), 1134-1140.

[57] Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical Bayesian models. *Journal of Mathematical Psychology*, 55(1), 1-7.

[58] Leslie, A. M. (1984). Infant perception of a manual pick-up event. *British Journal of Developmental Psychology*, 2(1), 19-32.

[59] Leslie, A. M., & Keeble, S. (1987). Do six-month-old infants perceive causality?. *Cognition*, 25(3), 265-288.

[60] Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 1-85.

[61] Low, J., & Watts, J. (2013). Attributing false beliefs about object identity reveals a signature blind spot in humans' efficient mind-reading system. *Psychological Science*, 24(3), 305-311.

[62] Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information.

[63] Masangkay, Z. S., McCluskey, K. A., McIntyre, C. W., Sims-Knight, J., Vaughn, B. E., & Flavell, J. H. (1974). The early development of inferences about the visual percepts of others. *Child development*, 357-366.

[64] Muentener, P., & Carey, S. (2010). Infants' causal representations of state change events. *Cognitive psychology*, 61(2), 63-86.

[65] Meltzoff, A. N., & Brooks, R. (2008). Self-experience as a mechanism for learning about others: a training study in social cognition. *Developmental psychology*, 44(5), 1257.

[66] Myung, I. J., & Pitt, M. A. (1997). Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic bulletin & review*, 4(1), 79-95.

[67] Nakahashi, R., Baker, C. L., & Tenenbaum, J. B. (2016, March). Modeling human understanding of complex intentional action with a bayesian nonparametric subgoal model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[68] Nakamura, E., Hamanaka, M., Hirata, K., & Yoshii, K. (2016, March). Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In 2016*IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 276-280). IEEE.

[69] Oakes, L. M. (2010). Using habituation of looking time to assess mental processes in infancy. *Journal of Cognition and Development*, 11(3), 255-268.

[70] Oakes, L. M., & Cohen, L. B. (1990). Infant perception of a causal event. *Cognitive Development*, 5(2), 193-207.

[71] O'Neill, D. K. (1996). Two-year-old children's sensitivity to a parent's knowledge state when making requests. *Child development*, 67(2), 659-677.

[72] Onishi, K. H., & Baillargeon, R. (2005). Do 15-month-old infants understand false beliefs? *Science*, 308(5719), 255-258.

[73] Paulus, M., Hunnius, S., van Wijngaarden, C., Vrins, S., van Rooij, I., & Bekkering, H. (2011). The role of frequency information and teleological reasoning in infants' and adults' action prediction. *Developmental psychology*, 47(4), 976.

[74] Penn, D. C., Holyoak, K. J., & Povinelli, D. J. (2008). Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences*, 31(2), 109-130.

[75] Perner, J., Mauer, M. C., & Hildenbrand, M. (2011). Identity: Key to children's understanding of belief. *Science*, 333(6041), 474-477.

[76] Phillips, A. T., & Wellman, H. M. (2005). Infants' understanding of object-directed action. *Cognition*, 98(2), 137-155.

[77] Piaget, J. (1964). Cognitive development in children. *Journal of Research in Science Teaching*, 2(2), 176-186.

[78] Quinn, P. C., Yahr, J., Kuhn, A., Slater, A. M., & Pascalis, O. (2002). Representation of the gender of human faces by infants: A preference for female. *Perception*, 31(9), 1109-1121.

[79] Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S. M., & Botvinick, M. (2018). Machine theory of mind. arXiv preprint arXiv:1802.07740.

[80] Raschle, N., Zuk, J., Ortiz-Mantilla, S., Sliva, D. D., Franceschi, A., Grant, P. E., & Gaab, N. (2012). Pediatric neuroimaging in early childhood and infancy: challenges and practical guidelines. *Annals of the New York Academy of Sciences*, 1252, 43.

[81] Repacholi, B. M., & Gopnik, A. (1997). Early reasoning about desires: evidence from 14-and 18-month-olds. *Developmental psychology*, 33(1), 12.

[82] Rasmussen, C. E. (2000). The infinite Gaussian mixture model. In *Advances in neural information processing systems* (pp. 554-560).

[83] Saparov, A., & Mitchell, T. M. (2016). A probabilistic generative grammar for semantic parsing. arXiv preprint arXiv:1606.06361.

[84] Schult, C. A. (2002). Children's understanding of the distinction between intentions and desires. *Child Development*, 73(6), 1727-1747.

[85] Schulz, L. E., Hooppell, C., & Jenkins, A. C. (2008). Judicious imitation: Children differentially imitate deterministically and probabilistically effective actions. *Child Development*, 79(2), 395-410.

[86] Schulz, L. E., & Sommerville, J. (2006). God does not play dice: Causal determinism and preschoolers' causal inferences. *Child development*, 77(2), 427-442.

[87] Segers, J. E. (1936). Recent observations relative to the perception of color in babies. *Archives Belges de Sciences et Education*, 2, 52-56.

[88] Searle, J. R., & Willis, S. (1983). Intentionality: An essay in the philosophy of mind. *Cambridge university press.*

[89] Sirois, S., & Mareschal, D. (2002). Models of habituation in infancy. *Trends in Cognitive Sciences*, 6(7), 293-298.

[90] Sirois, S., & Mareschal, D. (2004). An interacting systems model of infant habituation. *Journal of cognitive neuroscience*, 16(8), 1352-1362.

[91] Shahaeian, A., Peterson, C. C., Slaughter, V., & Wellman, H. M. (2011). Culture and the sequence of steps in theory of mind development. *Developmental psychology*, 47(5), 1239.

[92] Skinner, B. F. (1977). Why I am not a cognitive psychologist. *Behaviorism*, 5(2), 1-10.

[93] Sokolov, E. N. (1963). Higher nervous functions: The orienting reflex. *Annual review of physiology*, 25(1), 545-580.

[94] Spelke, E. S., Katz, G., Purcell, S. E., Ehrlich, S. M., & Breinlinger, K. (1994). Early knowledge of object motion: Continuity and inertia. *Cognition*, 51(2), 131-176.

[95] 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(2), 113-142.

[96] Spelke, E. S., & Kinzler, K. D. (2007). Core knowledge. *Developmental science*, 10(1), 89-96.

[97] Strehl, A. L., Li, L., Wiewiora, E., Langford, J., & Littman, M. L. (2006, June). PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning* (pp. 881-888).

[98] Stuhlmüller, A., & Goodman, N. D. (2014). Reasoning about reasoning by nested conditioning: Modeling theory of mind with probabilistic programs. *Cognitive Systems Research*, 28, 80-99.

[99] Surian, L., Caldi, S., & Sperber, D. (2007). Attribution of beliefs by 13-month-old infants. *Psychological science*, 18(7), 580-586.

[100] Teller, D. Y. (1984). Linking propositions. *Vision research*, 24(10), 1233-1246.

[101] Thomas, H., & Gilmore, R. O. (2004). Habituation assessment in infancy. *Psychological Methods*, 9(1), 70.

[102] Tomasello, M., & Haberl, K. (2003). Understanding attention: 12-and 18-month-olds know what is new for other persons. *Developmental psychology*, 39(5), 906.

[103] Thompson, R. F., & Spencer, W. A. (1966). Habituation: a model phenomenon for the study of neuronal substrates of behavior. *Psychological review*, 73(1), 16.

[104] Valentine, C. W. (1913). Colour perception and preferences of an infant at three months. *Section on Ophthalmology*, 18, 689-690.

[105] Weiss, Y., & Adelson, E. H. (1998). Slow and smooth: A Bayesian theory for the combination of local motion signals in human vision. Technical report A.I. Memo No. 1624, MIT, 1998.

[106] Wellman, H. M., & Bartsch, K. (1988). Young children's reasoning about beliefs. *Cognition*, 30(3), 239-277.

[107] Wellman, H. M., & Liu, D. (2004). Scaling of theory-of-mind tasks. *Child development*, 75(2), 523-541.

[108] Wellman, H. M., & Woolley, J. D. (1990). From simple desires to ordinary beliefs: The early development of everyday psychology. *Cognition*, 35(3), 245-275.

[109] Westfall, P. H. (2014). Kurtosis as peakedness, 1905–2014. RIP. *The American Statistician*, 68(3), 191-195.

[110] Wimmer, H., & Perner, J. (1983). Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1), 103-128.

[111] Woodward, A. L. (1998). Infants selectively encode the goal object of an actor's reach. *Cognition*, 69(1), 1-34.

[112] Woodward, A., Phillips, A., & Spelke, E. S. (1993). Infants' expectations about the motions of inanimate vs. animate objects. In *Proceedings of the Cognitive Science Society*. Erlbaum.

Appendix A: Overview of WebPPL

In this appendix, we present the basics of probabilistic programming languages (PPLs), and relevant details for the particular language we shall use (WebPPL).

A1: Basic syntax

WebPPL is an extension of a functional subset of Javascript, and therefore inherits much of the Javascript syntax.

- Objects: An object is defined as a set of features, e.g.

  var $myObject$=$\{feature_1 : value_1, feature_2 : value_2, ...\}$

  The value of each feature can be accessed using $myObject.feature_i$.

- Functions: New functions are defined using the syntax

  var $myFunction$ =function($inputs$){

  $\vdots$

  (code)

  $\vdots$

  return $output$ }

  As a functional programming language, each computation must be within the scope of some return statement.

- Conditionals: In WebPPL, the syntax $A\ ?\ B : C$ reads as "if A then B otherwise C." This allows for compact conditional return statements, e.g.

  var $myFunction$ =function(x){

  return $(x > 2)\ ?\ true : false$}

  which returns $true$ if $x > 2$ and $false$ otherwise.

A2: Probabilistic programs and stochastic computation

Stochastic primitives and stochastic recursion

WebPPL extends a functional subset of Javascript with stochastic primitive functions. For example, the primitive function $flip(w)$ returns a sample from a Bernoulli distribution with weight parameter w (i.e. 1 with probability w or 0 with probability $1 - w$). As a functional programming language, WebPPL does not support looping statements (e.g. for, $while$), but these computations can still be performed using recursion. For example, consider the following piece of code:

```
var geometric = function(w) {
    return (flip(w) ? 1 : 1 + geometric(w))
}
```

A call to $geometric(w)$ returns 1 with probability w or $1 + geometric(w)$ with probability $1 - w$, so this procedure corresponds to flipping a coin with weight w until the coin lands on heads, then returning the number of flips. Thus, this program explicitly defines a stochastic procedure for generating samples from a geometric distribution with parameter w.

Distribution objects

A probabilistic program like the previous example defines an explicit stochastic procedure for generating samples. However, such a program also implicitly defines a probability distribution over return values (conditioned on input values). In WebPPL, we can construct these implicit distributions as explicit distribution objects using a higher-order "marginalization operator," which transforms a sampling procedure into a corresponding distribution object. This is done using the Infer operator, which has the following syntax:

```
var myDist=Infer({model : f, options : infer_opts})
```

Here, the model f is any nullary function, and $infer\_opts$ specifies a method for constructing the marginal distribution. WebPPL supports several different inference methods, which we discuss in section A4 (though we ignore these details for now). For example, we can rewrite our $geometric(w)$ function within the scope of an Infer statement as

```
var geometric_dist(w) = function(w){
    return Infer(function(){
        return geometric(w)
    })
}
var myDist = geometric_dist(w)
```

A call to $geometric\_dist(w)$ returns a distribution object encoding the distribution over return values of the sampling procedure $geometric(w)$. We can then manipulate this distribution object using built-in functions such as

- sample($myDist$), which draws a single sample from $myDist$,
- $myDist$.score(x), which returns the score (negative log-probability) of value x under the distribution $myDist$,
- viz.table($myDist$), which generates a table of the joint distribution $myDist$.

In addition to the Infer operator, WebPPL also includes primitive distribution objects corresponding to each stochastic primitive. For each primitive, the corresponding primitive distribution shares the same name with a capitalized first letter. For example, the stochastic primitive $categorical(\{ps : [.3, .3, .4], vs : [0, 1, 2]\})$ returns a sample from a categorical distribution over the values $[0, 1, 2]$ with parameters $[.3, .3, .4]$, while $Categorical(\{ps : [.3, .3, .4], vs : [0, 1, 2]\})$ is an object corresponding to the categorical distribution itself.

Generative models in WebPPL

WebPPL is well suited for efficiently coding generative models. To illustrate, consider the following frequently used example of a causal system:

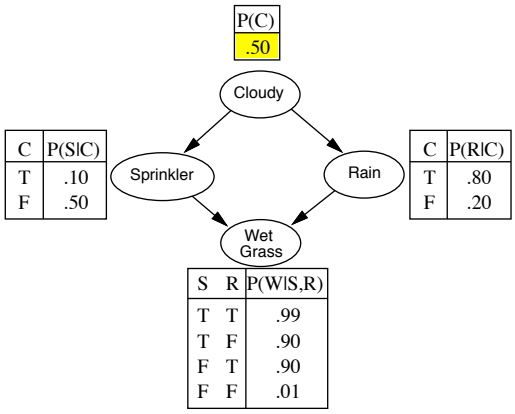


Figure 6.2.1: Common example of a causal system

This model defines a joint probability distribution over the variables C, S, R, W, which factors as $P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R)$. In WebPPL, we can encode this model as the following program:

```
var model = function() {
    var C = flip(.5)
    var S = C ? flip(.1) : flip(.5)
    var R = C ? flip(.8) : flip(.2)
    var W = (S&&R) ? flip(.99) : [(S||R) ? flip(.9) : flip(.01)]
    return [C, S, R, W]
}
```

Here, && and || denote the *and* and *or* operators, respectively, so the line

$$W = (S\&\&R) \ ? \ flip(.99) : \ [(S||R) \ ? \ flip(.9) : \ flip(.01)]$$

returns $flip(.99)$ if S and R are both true, $flip(.9)$ if one of S or R are true, and $flip(.01)$ if neither S nor R are true. Thus, the conditional probability tables in the above

diagram can be encoded with a set of conditional coin-flips, conditioned on the values of the corresponding parent variables. Given this program, the command $model()$ returns a sample of $[C, S, R, W]$ drawn from the distribution implied by this sampling procedure, which corresponds to the distribution shown in the causal diagram of the system.

Note that, like the stochastic primitive $flip()$, the program $model()$ defined above is a *procedure* for drawing samples from the distribution $P(C, S, R, W)$, but the distribution itself is only implicit in the return values of the program. However, we can once again use the Infer operator to transform this procedure into an explicit marginal distribution object. This is done as follows:

```
var distribution = function(){
    var model = function(){
        var C = flip(.5)
        var S = C ? flip(.1) : flip(.5)
        var R = C ? flip(.8) : flip(.2)
        var W = (S&&R) ? flip(.99) : [(S||R) ? flip(.9) : flip(.01)]
        return [C, S, R, W]
    }
    return Infer(model())
}
```

In this case, the command $distribution()$ returns the distribution object itself, and we can then draw samples from this distribution using $sample(distribution())$, or compute the log-probability of observing a particular draw $[C_0, S_0, R_0, W_0]$ using $distribution().score([C_0, S_0, R_0, W_0])$.

Queries and conditions

The program *distribution*() defined above returns a joint probability distribution over the four variables C, S, R, W. However, probabilistic graphical models are often used to infer the values of some subset of variables (or compute the marginal probabilities of those variables), given information about the values of other variables in the system. For example, we may observe that the grass is currently not wet, and wonder whether the sprinkler is on. WebPPL provides several higher-order functions for efficiently incorporating observations into the marginal distribution. Suppose, following our example, that we observe $W = false$, and we wish to compute the probability that the sprinkler is on $P(S = true|W = false)$. We can do this by modifying the above code as follows:

```
var distribution = function(){
    var model = function(){
        var C = flip(.5)
        var S = C ? flip(.1) : flip(.5)
        var R = C ? flip(.8) : flip(.2)
        var W = (S&&R) ? flip(.99) : [(S||R) ? flip(.9) : flip(.01)]
        Condition(W == false)
        return S
    }
    return Infer(model())
}
```

Note the two changes to this program: first, we add the line Condition($W == false$), which tells the $Infer$ operator that the distribution is conditioned on W taking on a particular value. Second, we change the return statement, so that we only return the variable in which we are interested. Thus, the modified code returns a distribution

object corresponding to $P(S|W = false)$.

The Condition operator works well for simple discrete observations, but WebPPL has two other built-in operators that allow for conditioning on observations. The first is Observe($dist, val$), where $dist$ is a distribution object and val is the observed value of a sample from that distribution. For example, if the command

$$\text{Observe}(Gaussian(\{mu\text{: } 0, sigma\text{: } 1), .01)$$

is within the scope of an Infer statement, the resulting distribution will be conditioned on observing the value .01 drawn from a standard normal distribution.

The third built-in method for conditioning on observations is the Factor operator, where Factor(v) adds the value v to the log-likelihood of the current iteration. Factor can also be used with conditional statements, e.g.

$$\text{Factor}(condition\ ?\ true\_factor\ :false\_factor)$$

This operator allows for "soft" conditioning: if the above command is within the scope of an Infer operator, then in each execution, Infer will add either $true\_factor$ or $false\_factor$ to the negative log-probability of the current execution, depending on whether *condition* is satisfied in that execution. Technically, Observe and Condition are both written in terms of Factor: Observe($dist, val$) is (usually[69]) equivalent to Factor($dist.score(val)$), and Condition(A) (where A is some binary valued statement) is

[69]certain Infer methods may process an Observe command in a more efficient way, but this will not be relevant in the current project

equivalent to Factor(A ? 0 : $-Infinity$). In many cases, however, Observe and Condition allow for more efficient notation.

In general, suppose we have a program var *model* = function() { ... }, which defines a generative model over some set of variables $X_1, \ldots, X_n$. If we let *query* be any subset of these variables, and *conditions* be a list of ordered pairs of the form $(X_i, val(X_i))$ (i.e. a variable and an observed value of that variable), we can define a function that returns the marginal distribution over the variables in *query*, given the observations in *conditions*:

```
var query_model = function(query, conditions) {
    var model = function() {
        (... model specifications ...)
        Factor(X_{i_1} == val(X_{i_1})&& ... &&X_{i_m} == val(X_{i_m})) ? 0 : -Infinity)
        return query
    }
    return Infer({model()})
}
```

The conjunctive conditional statement inside Factor checks whether the observations in *conditions* match the values of the corresponding variables in the current execution. Thus, this program returns the distribution $P(queries|conditions)$, i.e. the joint distribution over variables in *queries*, conditioned on the observations in *conditions*. This modularity allows us to efficiently code marginal distributions for arbitrary *query*/*conditions* combinations using a single model definition.[70]

[70] As we explain in the next section, this general form will not always be tractable, but it at least demonstrates that the function for computing these conditional distributions is well defined at the computational level

A3: Interlude on inference algorithms

The previous sections provide a computational-level description of the WebPPL inference functions we will use. However, as this project explores issues at both the computational and algorithmic levels, it is important to have a basic understanding of the actual inference methods. We shall describe the three main inference methods we shall use here, so that we can later explore how these algorithms constrain the space of problems that the inference agent can tractably solve.

Rejection sampling

The most conceptually straightforward inference method is rejection sampling, which has syntax Infer({method: *rejection*, samples: *numSamples*}). With these options, the Infer operator repeatedly runs the generative model and rejects any sample which conflicts with a Condition statement (or any sample which includes a Factor($-Infinity$)) statement, until *numSamples* samples are accepted. The operator then returns the empirical distribution over accepted samples, weighted by any additional Factor scores.

While rejection sampling is conceptually straightforward, its efficiency depends on the prior probability of the conditioning statements. Suppose our model includes a single Condition(A) statement, and that the prior probability that a given sample will satisfy A is p. The number of samples required to obtain a single compatible sample follows a geometric(p) distribution, so the average number of samples required to obtain n successes is n/p. If p is very low, this number can be prohibitively high, rendering rejection sampling intractable for cases where the prior probability of a condition is very low (e.g. if we condition the data on a single possible trial out of a very large number of

possible alternatives). This will rule out rejection sampling for most of our programs, but it is still conceptually important to understand.

Enumeration

Unlike sampling based methods, enumeration explicitly computes the probability of every possible path in the model's execution, weighting each path by the total Factor scores incurred within that execution. This method has options $\{maxExecutions, strategy\}$. The first option controls the maximum number of (complete) executions to enumerate (the default is Infinity). The second option controls the traversal strategy for enumeration ('likely-first,' 'depth-first,' or 'breadth-first'). The efficiency of this method depends on the total number of possible execution histories compatible with the conditions, rather than the prior probability of those conditions. Enumerate is therefore inapplicable to distributions involving continuous parameters, which entail an infinite number of possible execution histories.

MCMC

This method performs a Markov Chain Monte Carlo (MCMC) random walk through the posterior and returns the empirical distribution over points visited in this walk (weighted, again, by appropriate Factor scores). The options for this method control the total number of samples to propose, the burn-in (number of initial samples to discard from the final distribution), lag (number of iterations to perform between samples), and transition kernel. With the default transition kernel, the MCMC method performs a standard Metropolis-Hastings (MH) algorithm, though WebPPL includes several built-in kernels for MCMC inference (e.g. Hamiltonian Monte Carlo). Compared to rejection

sampling, MCMC methods are significantly more robust to increases in the number of possible execution histories and decreases in the prior probability of conditions (although the algorithm still relies on rejection sampling to generate an initial sample with nonzero posterior probability). However, because of the autocorrelation between consecutive samples, the default MH algorithm may become "stuck" in certain cases, where it is difficult to generate new samples with nonzero posterior probability, given the current position of the sampler. This will become relevant as we get into level 2 inference programs.

A4: Memoization and parameter inference

In computer science, memoization is a technique used to avoid repeating computations that have already been performed. WebPPL includes a higher-order memoization function $mem()$, which, when applied to a function $f(inputs)$, returns a memo-ized version of that function. The first time $mem(f(x))$ is called with a given argument x, mem will compute and return the value $f(x)$; on subsequent calls with the same argument value, mem will bypass the full computation of $f(x)$, and instead return the value that was previously computed with the same arguments.

If f is a deterministic program, $mem(f)$ represents the exact same input-output map as f; the former is simply less expensive to compute. If f is a stochastic function, however, $mem(f)$ will generally have a very different meaning (distribution over return values) than f. For example, if we call $flip(.5)$ multiple times, each call has an equal probability of returning the values $true$ or $false$. If we call $mem(flip(.5))$ several times, the first call has an equal probability of returning $true$ or $false$, but each subsequent call

will return the same value as the first call with probability 1.

The *mem* function is useful for generating persistently random features in generative models. Suppose, for example, we wish to describe each agent in a model as either left- or right-handed. A simple program for randomly determining this attribute would be

```
var handedness = function(person) {
    return flip(.1) ? 'L' : 'R'
}
```

If we apply this function to each element in the list $[bob, alice, charles]$, it would output a "handedness" value to each person. If, however, we called $handedness(bob)$ a second time, it would not necessarily return the same label as the first call, since each call to $handedness(bob)$ invokes $flip(.1)$, which has a non-deterministic output. However, suppose we memoize this function:

```
var handedness = mem(function(person) {
    return flip(.1) ? 'L' : 'R'
)}
```

In this case, the first call to $handedness(bob)$ would assign a label to *bob*, and each subsequent call with the same argument would return the same label. Note that the argument *person* isn't actually needed to compute the return value of the non-memoized version of *handedness*; in this case, the argument is used only to index the return values of the memoized function.

This technique is useful for models in which we have multiple parameters drawn from the same prior distribution. For example, consider a scenario in which we have several bags of marbles, each containing some distribution of five different colors of marbles. A

single draw from each bag can be represented as a categorical distribution over five possible outcomes, with one parameter for each color. We can use the *mem* operator to define a program that generates a categorical distribution for each bag:

```
var makeBag = mem(function(bagname) {
    return Categorical({
        ps: T.toScalars(dirichlet(ones([5, 1]))),
        vs: [blue, yellow, red, green, purple]})
})
```

Here, each parameter set for each Categorical distribution is drawn from a symmetric Dirichlet prior with concentration parameter $\alpha = 1$. By using the *mem* operator, the first call to $makeBag('bagN')$ will generate a categorical distribution over colors for bagN, and each subsequent call with input value bagN will return the same distribution.

Suppose we observe multiple draws from two different bags, and want to infer the distribution of marbles inside each bag (i.e. parameters of the corresponding categorical distribution). We can represent the results of learning in terms of a posterior predictive distribution, i.e. a single hypothetical draw from each bag. We can do this using the *observe* operator in conjunction with our memoized bag-generating function:

```
var marble_distribution = Infer(function() {
    var bag1 = makeBag(bag1)
    var bag2 = makeBag(bag2)
    var data1 = [blue, blue, green, blue, blue, red, blue, purple, blue]
    var data2 = [yellow, blue, blue, blue, green, blue, purple]
    var observe_data = function(datum, bagname) {
        observe(makeBag(bagname), datum) }
    mapData({data: data1}, observe_data)
    mapData({data: data2}, observe_data)
    return {bag1: sample(makeBag(bag1)), bag2: sample(makeBag(bag2)) }
})
viz.marginals(marble_distribution)
```

Recall that the observe(*dist*, *val*) function adds the condition that the value *val* was drawn from distribution *dist*. The mapData(*data*, *function*(*d*)) function returns the array obtained by applying *function*(*d*) to each element in the array *data*, so these two lines of code specify that the sequences *data*1 and *data*2 were drawn from the distributions *bag*1 and *bag*2, respectively. Since each *bagN* is a categorical distribution object, *sample*(*makeBag*(*bagN*)) returns a single draw from this distribution. Thus, the function defined inside *Infer* returns a single hypothetical draw from each bag, conditioned on observing the sequences *data*1 and *data*2 drawn from each bag, and *marble_distribution* is the object containing the predictive posterior distribution for each bag. The last line of code allows us to visualize each marginal distribution in *marble_distribution* as a histogram of draws from that bag.

A5: Hierarchical models

WebPPL is well-suited for modular coding of complex hierarchical models. To illustrate this, consider our bag-of-marbles example in the previous section. We have already

shown how to infer the distribution of marbles inside a bag, given some observations (a sequence of draws) from that bag. Suppose now that we observe a sequence of draws from each of three bags, and we wish to generalize this knowledge to a fourth bag, from which we have observed no draws. That is, given the inferred distribution of marbles inside each of the first three bags, what would we expect the distribution of marbles to be in a fourth bag that we haven't observed at all?

We can modify the program in the previous section so that it a) includes a sequence of observations from a third bag and b) returns the predictive posteriors of the first three bags as well as a fourth, hypothetical bag:

```
var marble_distribution = Infer(function() {
    var bag1 = makeBag(bag1)
    var bag2 = makeBag(bag2)
    var bag3 = makeBag(bag3)
    var bag4 = makeBag(bag3)

    var data1 = [blue, yellow, green, blue, blue, red, blue, purple, blue]
    var data2 = [yellow, blue, blue, blue, green, blue, purple]
    var data3 = [blue, blue, red, blue, green, blue, blue]

    var observe_data = function(datum, bagname) {
        observe(makeBag(bagname), datum) }

    mapData({data: data1}, observe_data)
    mapData({data: data2}, observe_data)
    mapData({data: data3}, observe_data)

    return {bag1: sample(makeBag(bag1)),
        bag2: sample(makeBag(bag2)),
        bag3: sample(makeBag(bag3)),
        bag4: sample(makeBag(bag4))},
})
viz.marginals(marble_distribution)
```

In this case, the last line of code will generate a histogram of hypothetical draws from each of the first three bags, as well as a histogram of hypothetical draws from a hypothetical fourth bag from which we have observed no draws. However, given the above model, we should expect that, with enough draws, the histogram of the fourth bag will be nearly uniform (and indeed it is). This is because the model in this program learns a separate mixture for each bag, but does not learn any higher-level prototype or common information about all four bags in general. This is akin to learning a prototype for each of the classes *sedan*, *convertible*, *SUV*, etc. without learning the higher-level prototype *automobile*. Thus, regardless of what we observe from the first three bags, we should expect this model to produce a roughly uniform estimate of the marble mixture in the fourth bag.

In order to perform the sort of generalization required here, we need to add an additional level of abstraction to our generative model. Consider the following modification of our program:

```
var marble_distribution = Infer(function() {

    var prototype = T.mul(dirichlet(ones([5, 1])), 5)

    var makeBag = mem(function(bagname) {
        return Categorical({
            ps: T.toScalars(dirichlet(prototype)),
            vs: [blue, yellow, red, green, purple]})
    })
    var bag1 = makeBag(bag1)
    var bag2 = makeBag(bag2)
    var bag3 = makeBag(bag3)
    var bag4 = makeBag(bag3)

    var data1 = [blue, yellow, green, blue, blue, red, blue, purple, blue]
    var data2 = [yellow, blue, blue, blue, green, blue, purple]
    var data3 = [blue, blue, red, blue, green, blue, blue]

    var observe_data = function(datum, bagname) {
        observe(makeBag(bagname), datum) }

    mapData({data: data1}, observe_data)
    mapData({data: data2}, observe_data)
    mapData({data: data3}, observe_data)

    return {bag1: sample(makeBag(bag1)),
        bag2: sample(makeBag(bag2)),
        bag3: sample(makeBag(bag3)),
        bag4: sample(makeBag(bag4))}
})
viz.marginals(marble_distribution)
```

This model first generates a single "prototype" draw from a scaled Dirichlet distribution, which is then used as a hyperparameter in the Dirichlet-Categorical distribution used to model each bag. Thus, the concentration parameter is now a learnable value, rather than a fixed vector. Because the fourth bag is generated using this

same prototype, the model allows us to generalize our observations about the first three bags to the fourth, even though we have not observed any draws from the fourth bag. Running this program returns a histogram for the fourth bag that is heavily concentrated on blue, and roughly uniform among other colors (as is consistent with our observations of the first three bags). Furthermore, the extra level in this model enables faster learning about the first three bags, as compared to the previous version with no shared prototype.

It is worth noting that that the conditioning and return statements in this program are left unchanged. WebPPL allows us to add an extra hierarchical level to the model without having to re-specify the whole inference procedure. This enables efficient, modular construction of some very complex hierarchical distributions, and allows us to represent learning at a very general level.

Appendix B: implementation of the computational framework

Here we present the code and details for our implementation of the computational framework described in chapter 3. Recall that Level 1 problems involve inference about a single episode of a single actor's behavior, Level 2 problems involve inference about a single actor from one or more episodes of that actor's behavior, and Level 3 problems involve inference about actors in general from multiple episodes of multiple actors' behavior.

B1: Level 1

Data generation

A level 1 problem involves inference about a particular actor within a particular trial. The complete data D for a single trial consist of two features, $D.states = [s_0, s_1, \ldots, s_t]$ and $D.acts = [a_0, a_1, \ldots, a_{t-1}]$. For the first set of examples we will use a set of Gridworld environments in which each state has two features $s.grid$ and $s.inv$. $s.grid$ is an $n \times m$ character grid, and the contents of cell i, j are determined by the character(s) in that cell:

- 'E': empty (an actor can occupy and traverse this cell)
- 'W': wall (an actor can neither occupy nor traverse this cell)
- 'X': actor
- (any other character): other objects

The other feature $s.inv$ denotes the actor's "inventory;" objects can be added to this inventory via the "pickup" action.

The dynamics of this gridworld environment are defined by two core functions: $states\_toActs : S \to A$, which maps a system state to the set of allowable actions from that state; and a state-transition function $T : S \times A \to S$, which maps a current state and action to the next state.[71] For these gridworlds, we use the action set

[71] If the state-transition function is deterministic, which it will be for most of our simulations, we can technically do away with the action variable and transition function, instead coding the agent's action policy to directly output the next state. For our purposes it will be notationally more convenient to retain the action variable as a distinct entity

$A = \{0, 1, 2, 3, 4, 5\}$, where $0, 1, 2, 3$ correspond to movement by 1 step up, down, left, or right, respectively. 4 corresponds to the "pickup" action: if the actor shares a location with any object O (represented as ('X','O') in the corresponding cell), then selecting action 4 will add 'O' to the actor's inventory (and remove it from that cell). The last action 5 corresponds to ending the trial; this will generally occur when either the actor's goal is satisfied, or the actor exceeds a "maximum number of actions" cap, which we encode as a parameter of the MDP system. These changes are encoded by the state transition function T. The MDP itself is encoded as a System object, with features System.s2A (*states_toActs*), System.T(ransition function), and System.P(arameters).

To generate trial data, we need a "policy" to simulate an actor making sequential decisions in one of our gridworlds. In the next section, we will explain how we can use these policies to represent an observer's hypothesis for explaining an actor's behavior. For the purpose of *generating* the data, however, we need only think of this policy as a function $p : S \to A$ from states to (distributions over) actions. We impose the requirement that p assigns at least one state in S to the terminal action 5 (though our maximum action cap guarantees that the trial will halt even if all terminal states are unattainable from the initial configuration of the system). We can then generate our trial data with the following function:

```
var generate_trial = function(system, policy){
    var iterate = function(states, acts){
        var current_state = states[states.length − 1]
        var next_act = sample(policy(current_state))
        if (next_act == 5||acts.length > system.cap)
        {return {states : states, acts : acts.concat(next_act)}}
    else
        var next_state = system.T(current_state, next_act)
        return iterate(states.concat(next_state), acts.concat(next_act))}
    return iterate([system.init], [])}
```

The inner recursive function *iterate* takes a partial trial (sequence of n states and $n-1$ actions) and generates the next observation. If this observation is a terminal action (or the number of actions exceeds the system cap), the function halts and returns the current trial, otherwise it calls *iterate* again with the extended state list and act list. Thus, the output of $var\ D = generate\_trial(system, policy)$ is an object D with two features: $D.states$ (an array of state values) and $D.acts$ (an array of equal length containing action values). This object will serve as one of four inputs to the main inference function, explained in the next section.

Inference

Level 1 problems involve inference and prediction about a particular actor in a particular trial. This includes predicting future actions from initial observations, inferring the values of mental states from observed actions (and prior knowledge about the actor), counterfactual inference, and control-based reasoning. We will focus on action prediction and mental inference here, but the other two types of reasoning involve the same underlying computations over different inputs.

In order to maximize the modularity of these programs, we define all forms of Level 1 inference using a single function with the following syntax:

$$trial\_inference = \text{ function } (system, conditions, queries, actor\_model)$$

Each of these inputs is defined as follows:

- $system = \{T\text{: } transition, A\text{: } states\_toActs, P\text{: } parameters\}$, where $A(s)$ returns the set of viable actions from system-state s. For our first examples, $parameters$ consists of the single parameter *cap*, which imposes an upper limit on the number of actions an agent may take in a trial.

- $conditions = \{stateInfo\text{: } [[t^s_1, s_{t^s_1}], \ldots], actInfo\text{: } [[t^a_1, a_{t^a_1}], \ldots]\}$. This input encodes a partial observation of a trial, consisting of state observations and act observations. The index t^v_i denotes the ith observation of variable v (note that t^v_i and t^v_{i+1} are not necessarily consecutive), and $v_{t^v_i}$ denotes the observed value of the corresponding variable.

- $queries = \{states : [r^s_1, \ldots], acts : [r^a_1, \ldots], mental : [[r^m_1, \text{type}_1] \ldots]\}$. The *queries* object is similar to the *conditions* object, except that it contains only index numbers for each variable, and it may contain index numbers corresponding to mental variables as well. This input denotes the information that is being requested (i.e. the variables over which the output distribution is defined, conditioned on the observations in *conditions*).

- $actor\_model = \{af : action\_function, uf : update\_function, p : params\}$. This input defines the observer's model of a particular actor. The first two features are

an action function $af(m, s; p_a) \rightarrow a$ which maps a system state s and a mental state m to a distribution over actions; and a mental update function $uf(m, s, p_u) \rightarrow m'$ which maps a (current) mental state m and (new) system state s to a distribution over (new) mental states.[72] Both of these functions rely on actor-specific parameter values θ_a and θ_u, respectively; these parameter values are stored in the *params* object, i.e.

$$params = \{param_1 : \theta_1, param_2 : \theta_2, \ldots\}$$

At the core of the $trial\_inference$ function is a recursive *iterate* function very similar to the one in *generate_trial*, but with three important differences. The first is that the state values in *states* are split into two objects: the observable system states *s_states* and the unobservable mental states *m_states*. Similarly, the $policy(state) \rightarrow action$ function is decomposed into two functions: $af(s\_state, m\_state) \rightarrow action$, which maps a mental state and a system state to an action distribution, and $uf(m\_state, s\_state) \rightarrow m\_state$,[73] which maps a mental state and a system state to a mental state distribution. Thus, these first two differences separate out the "mental" part of the inference problem from the "physical."

The third difference is that each execution of $iterate(s\_states, m\_states, acts)$ includes a Factor statement, which checks whether the most recently generated

[72] uf can also be used without a value for the m argument, in which case it generates an "initial" mental state for that actor, given the initial system-state

[73] We encode the system specifications as part of the system-state argument s, which allows us to define these functions in a system-general way

observation in $(s\_states, acts)$ matches or mismatches any observation in *conditions*. In particular, the helper function $check\_conditions(conditions, new\_ob)$ returns $true$ if either new_ob matches with an observation in *conditions*, or the value of new_ob is missing in *conditions*, and returns $false$ if new_ob conflicts with an observation in *conditions*. We then include the statement

$$\text{Factor}(check\_conditions(conditions, new\_ob)\ ?\ 0 :\ -Infinity)$$

inside the *iterate* function. The value $-Infinity$ defaults to the maximum allowable negative integer, so this effectively conditions the resulting distribution on those trials that are consistent with the observations in *conditions*. Furthermore, multiple Factor statements within the scope of the same Infer statement will stack additively. This allows us to more efficiently execute the Infer operation, as any execution will halt as soon as it first encounters a generated observation that conflicts with *conditions*.

With this intuitive overview established, the code for $trial\_inference$ is as follows:

```
var trial_inference = function (system, conditions, queries, actor_model){

    //load system definitions
    var T = system.transition
    var A = system.states_to_acts
    var cap = system.params

    //load actor_model
    var af = actor_model.af
    var uf = actor_model.uf
    var params = actor_model.paras

    //construct trial model
    var trial_model = function (){
        //inner recursive loop
        var iterate = function (s_states, m_states, acts){
            var current_s = s_states[s_states.length − 1]
            var current_m = m_states[m_states.length − 1]
            var next_a = sample(af((current_s, current_m, params)))
            var next_s = T(current_s, next_a)
            var new_ob = {index: s_states.length − 1, state: next_s, act: next_a}

            //reject samples that violate conditions
            Factor(check_conditions(conditions, new_ob) ? 0 : −Infinity)
            var next_m = sample(uf(current_m, next_s, params))
            if (next_a == 5 or acts.length >= system.cap)
                {return[s_states, m_states, acts].concat(new_ob, next_m)}
            else
                {return iterate([s_states, m_states, acts].concat(new_ob, next_m))}}
        var trial = iterate([system.init], [sample(uf(system.init, params))], [])
        return get_query_info(query, trial)}
    return Infer(trial_model, {method: 'enumerate'})}
```

The *trial_model* function first loads the parameter values from *actor_model*. It then uses the recursive *iterate* function to generate a simulated trial for an actor with *actor_model* in the input *system*. The helper function *get_query_info* extracts the data

requested in *query* from the simulated trial, and the Factor statements inside *iterate* add $-Infinity$ to the log probability of any simulated trial that conflicts with an observation in *conditions*, effectively dropping any such trial from the computation. Thus, the distribution returned by the final Infer statement corresponds to the conditional distribution $P(queries|conditions, actor\_model)$, which is precisely the information required for Level 1 inference.

Actor Models

An actor model consists of a parameterized action function
$af : (s\_state, w\_state; \theta_a) \rightarrow P(acts)$, a parameterized mental update function
$uf : (m\_state, s\_state; \theta_u) \rightarrow P(m\_states)$, and a set *params* of parameter values, one for each parameter in af and uf. In Levels 2 and 3, we will discuss how these actor models are themselves inferred from data. To better illustrate how actor models are coded, however, we present the programs that define the three example actor models shown in chapter 3.4.

Error-prone fixed-goal Under an error-prone fixed-goal model (figure 3.4.1), the actor's mental state consists of a single discrete "goal" variable which points to a particular object or feature in the current environment. The value of this goal variable is sampled at the start of a trial, from those features or objects which are attainable from the initial system state. The mental update function for this model is as follows:

```
var uf =function(s, m, β){
    if (m ==undefined){
        var targets = objects(s)
        var values = filterValues(β, targets)
        return Categorical({vals: targets, probs: softMax(values)})
    }
    else {returnDelta(m)}
}
```

The β parameter is an $n \times 1$ vector of non-negative reals, where β_i denotes the value that the actor would derive from obtaining object i. If the m argument is empty (i.e. when uf is first called to initialize the actor's mental state), the program identifies the set of objects attainable from the initial system state, and pulls out the value parameters associated with those objects. It then normalizes those values into a probability vector using a softMax transformation, and returns a categorical distribution over the set of attainable objects. Thus, the probability that the actor will target object i is directly dependent on the value that the actor would derive from obtaining object i. In each subsequent step, the function returns a Delta distribution concentrated on the previous goal value.

The action-selection function for this model is as follows:

```
var af =function(s, m, ϵ){
    var optMoves = pathFinder(s, m)
    var optProbs = repeat(optMoves.length, 1 − ϵ)
    var subMoves = remove(states_toActs(s), optMoves)
    var subProbs = repeat(subMoves.length, ϵ)
    return Categorical({vals: optMoves.concat(subMoves),
                        probs: optProbs.concat(subProbs)})
}
```

Under this program, the actor computes the shortest path(s) from their current location to the location of the goal object, and takes either a) a step along one of the shortest paths to the goal (sampled uniformly at random if there are multiple shortest paths) with probability $1 - \epsilon$, or b) a random "sub-optimal" step with probability ϵ. The structure of this actor model is defined by the two programs af and uf defined above, and the parameter space consists of the value parameter β and the error parameter ϵ.

Deterministic mind-change Under a deterministic mind-change model (figure 3.4.2), the actor's mental state consists of the same discrete "goal" variable as in the error-prone fixed-goal model. Unlike the previous model, the actor may change their mind in each step, with probability ν (the "fickleness" parameter). The mental update function for this model is as follows:

```
var uf =function(s, m, β, ν){
    if (m ==undefined||flip(ν)){
        var targets = objects(s)
        var values = filterValues(β, targets)
        return Categorical({vals: targets, probs: softMax(values)})
    }
    else {return Delta(m)}
}
```

The action-selection function for this model is similar to the function for the previous model, but will always return an optimal step towards the current target object with probability 1:

```
var af =function(s, m){
    var optMoves = pathFinder(s, m)
    return UniformDraw(optMoves)
}
```

Awareness-constrained rational Under an awareness-constrained model (figure 3.4.3), the actor's mental state consists of a discrete goal variable *and* an awareness variable, which consists of a list of features of which the actor is currently aware (and the location of each feature). The mental update function for this model first updates (or initializes) the actor's awareness state, then updates (or initializes) the actor's goal based on their new awareness state:

```
var uf =function(s, m, β, ρ){
    if (m ==undefined){
        var A = line_of_sight(s, ρ)
        var targets = objects(A)
        var values = filterValues(β, targets)
        var goalDist = (targets.length == 0)? Delta('search'):
                        Categorical({vals: targets, probs: softMax(values)})
        return {a: A, g: goalDist}
    }
    else{
        var newA = extend(m.A, line_of_sight(s, ρ))
        if (newA == m.A){return Delta(m)}
        else{
            var targets = objects(A)
            var values = filterValues(β, targets)
            var goalDist = (targets.length == 0)? Delta('search'):
                            Categorical({vals: targets, probs: softMax(values)})
            return {a: newA, g: goalDist}
        }
    }
}
```

This program first updates (or initializes) the actor's current awareness state by adding the ID and location of any new features detected via $line\_of\_sight(s, \rho)$. This helper function scans the actor's line of sight until it a) reaches the edge of the grid, b) hits a wall, or c) exceeds the actor's visual range, encoded by parameter ρ. If this

function detects any new objects, the actor resamples their goal state from the set of objects in their new awareness state. Otherwise, the function returns the same goal state. If the actor's current awareness state does not include any potential target object, the function sets the goal state equal to 'search,' which specifies that the actor should execute a search behavior. If the actor's current goal state is 'search,' the action-selection function outputs a random movement in any direction, other than a direct backtrack to a previously occupied location (unless there are no other possible moves). Otherwise, the function outputs an optimal step towards the current target object. This function is as follows:

```
var af =function(s, m){
    if(m.g ==='search'){
        var moves = states_toActs(s)
        return UniformDraw(moves)
    }
    else{
        var optMoves = pathFinder(s, m.g)
        return UniformDraw(optMoves)
    }
}
```

B2: Level 2

Data generation

The data for Level 2 problems involve multiple episodes of a single agent's behavior.[74] To quickly generate large data sets of episodes, we code, for each system, a *random_initialState*() function, which outputs a random initial state from the system according to some parameters. For the gridWorld system, these parameters are: grid dimensions (x- and y-), a list of objects that the grid may contain (*random_initialState* randomly selects at least 2 of the objects to include in random locations), and the number of walls in the grid (also randomly placed). We can quickly generate a large data set of trials for a particular actor model using the command

---

var $data = repeat(numTrials,$
function(){return $gen\_trial(actor\_model, random\_initialState())$})

Inference

Similar to Level 1 inference, we can perform Level 2 inference using a single functional form with the following syntax:

$$actor\_inference = \text{function}(trials, theory, query, inferOpts)$$

[74]Note that the actor model functions, trial generation function, and pathFinder function are coded in a system-general way. This allows the observer to do Level 2 inference over episodes collected from multiple different systems (assuming the observer knows *states2Acts* and *transition* for each system)

where

- $trials$ is an array $t_1, \ldots, t_n$ of trials involving a particular actor,
- $theory$ is the observer's *framework theory*, which encodes a prior distribution over actor models,
- $query$ specifies the requested information. This may include a full posterior distribution over actor models, or a posterior distribution over a particular component of an otherwise fixed actor model (e.g. one particular parameter or constraint), and
- $inferOpts$ specifies the method for the Infer operator. Because Level 2 inference often involves continuous parameters, this will be either $rejection$ or $MCMC$.

The observer's framework theory T consists of a set of model structures (each corresponding to a pair of program forms af and uf), a prior distribution $T.structurePrior$ over model structures, and a prior distribution $T.paramPrior[structure]$ over the parameter space for each structure. For certain inference problems, T will only specify a single model structure, in which case $T.structurePrior$ is a Delta distribution concentrated on that structure.

The most straightforward program for performing Level 2 inference is as follows:

```
var actor_inference =function(trials, theory, query, inferOpts){
    return Infer(inferOpts,function(){
        //sample actor model from theory prior
        var structure = sample(theory.structurePrior)
        var params = sample(theory.paramPrior[structure])
        var model = {af: structure.af, uf: structure.uf, params: params}
        //core function to observe trial data
        var observeTrial =function(trial){
            //load trial data
            var stateData = trial.states
            var actData = trial.acts
            //inner recursive loop to observe each step of trial
            var iterate =function(s_states, m_states, acts){
                if(s_states.length == stateData.length){ return true}
                else{
                    var current_s = s_states[s_states.length - 1]
                    var current_m = m_states[m_states.length - 1]
                    var next_a = actData[acts.length]
                    var actDist = model.af(current_s, current_m, model.params)
                    //add likelihood of observing next act
                    Observe(actDist, next_a)
                    var next_s = stateData[s_states.length]
                    var next_m = sample(model.uf(current_m, next_s, model.params))
                    return iterate(s_states.concat(next_s),
                                   m_states.concat(next_m),
                                   acts.concat(next_a))
                }
            }
            return iterate([stateData[0]], [], [])
        }
        var addFactors = map(observeTrial, trials)
        return getQueryInfo(model, query)
    })
}
```

The first part of this program samples a model structure, then samples a parameter vector for that structure. The *observeTrial* function iterates through each step of a

single trial and adds the likelihood of observing those actions under the sampled actor model. This requires sampling the actor's mental state at each step. The $map(observeTrial, trials)$ command applies the $observeTrial$ function to each trial in the data set $trials$. The last line extracts and returns the requested information from $model$. Thus, this program returns the posterior distribution $P(query|trials, theory)$.

While this program is theoretically sound, it will often be intractable outside of simple cases. The reason for this is that the MCMC method relies on rejection sampling to construct an initial sample with non-zero posterior probability. In order to construct an initial sample, the program must repeatedly execute every *sample* statement- including the initial two statements to sample an actor model, *and* each $next\_m = sample(model.uf(current\_m, next\_s, model.params))$ for each step of each trial- until it generates a sample with non-zero posterior probability. If the data contain a large number of trials, or there are a large number of possible alternative trials in each system, this may require a very large number of samples to generate an initial configuration with non-zero posterior probability. In order to make the program more tractable, we can replace the *sample* statement with a MaP estimate of the actor's mental state, using the Level 1 inference function $trial\_inference$ and a higher-order $Mode(dist)$ function, which returns the most likely value of a distribution $dist$. We then weight that execution with the likelihood of the MaP mental state. While this is only an approximation of the true posterior, the observer's bias for deterministic explanations means that this will generally be a reasonably accurate approximation. This significantly reduces the number of *sample* statements that need to be executed.[75] The code for this more efficient program is as follows:

[75]from $2 + k * n$, where $k * n$ is the total number of steps across all trials in $trials$, to 2

```
var actor_inference =function(trials, theory, query, inferOpts){
    return Infer(inferOpts,function(){
        //sample actor model from theory prior
        var structure = sample(theory.structurePrior)
        var params = sample(theory.paramPrior[structure])
        var model = {af: structure.af, uf: structure.uf, params: params}
        //core function to observe trial data
        var observeTrial =function(trial){
            //load trial data
            var stateData = trial.states
            var actData = trial.acts
            //inner recursive loop to observe each step of trial
            var iterate =function(s_states, m_states, acts){
                if(s_states.length == stateData.length){ return true}
                else{
                    var current_s = s_states[s_states.length - 1]
                    var current_m = m_states[m_states.length - 1]
                    var next_a = actData[acts.length]
                    var actDist = model.af(current_s, current_m, model.params)
                    //add likelihood of observing next act
                    Observe(actDist, next_a)
                    var next_s = stateData[s_states.length]
                    //compute MaP estimate of actor's next mental state
                    var next_m = Mode(trial_inference(trial, [m_state, m_states.length], model))
                    //add likelihood of MaP mental state
                    var m_dist = model.uf(next_s, current_m, model.params)
                    observe(m_dist, next_m)
                    return iterate(s_states.concat(next_s),
                                   m_states.concat(next_m),
                                   acts.concat(next_a))
                }
            }
            return iterate([stateData[0]], [], [])
        }
        var addFactors = map(observeTrial, trials)
        return getQueryInfo(model, query)
    })
}
```

B3: Level 3

Data generation

The data for Level 3 inference problems involve multiple episodes of multiple actor's behavior. To generate multiple trial observations for a single actor, we can use the procedure defined in the previous section. The method for generating a population of actors depends on the scope of the problem we are trying to model. For our purposes, we distinguish between two kinds of Level 3 data sets: single-population and mixed-population. In a single-population problem, we define a single prior distribution $P(\mathcal{M})$ over actor models, sample n actor models $\mathcal{M}_1, \ldots, \mathcal{M}_n$ from this prior, and then apply the Level 2 data generation procedure described in the previous section to each actor model, to obtain a data set consisting of k_i trials for each of the n actors. In a mixed-population problem, we divide the n agents into $m < n$ sub-populations, each corresponding to a separate prior $P_i(\mathcal{M})$ over actor models.

Inference

There are many forms that Level 3 inference can take, depending on the scope of the problem we are trying to model and the depth of the observer's background knowledge. We distinguish between 4 classes of Level 3 inference problem: single-population parameter inference, single-population full-model inference, mixed-population parameter inference, and mixed-population full-model inference. All four classes of inference can be performed using a single program form, but there are non-trivial differences between the helper programs used for each kind of inference. We first present the general program form, before explaining the differences.

The general Level 3 inference program has the following syntax:

$$theory\_inference = \text{function}(data, overhyp, inferOpts)$$

where

- $data = \{D_1, \ldots, D_n\}$ is an $n \times 1$ array of actor-specific data sets, and each D_i is an $n_i \times 1$ array of trial observations for actor i.
- $overhyp$ is the observer's over-hypothesis, which encodes a prior distribution over theories. The form of this over-hypothesis, and the helper function for sampling theories, is the main difference between the four forms of Level 3 inference.
- $inferOpts$ specifies the method for the Infer operator. Because of the scope of Level 3 inference problems, this will almost always be an MCMC variant.

The general form for this program is as follows:

```
var theory_inference =function(data, overhyp, inferOpts){
    return Infer(inferOpts, function(){
        var theory = sampleTheory(overhyp)
        var numActors = data.length
        var actorModels = mapN(function(n){return sampleModel(theory)},
                               numActors)
        var observeTrial =function(actorID, trial){
            var model = actorModels[actorID]
            var stateData = trial.states
            var actData = trial.acts
            var iterate =function(s_states, m_states, acts){
                if(s_states.length == stateData.length){ return true}
                else{
                    var current_s = s_states[s_states.length - 1]
                    var current_m = m_states[m_states.length - 1]
                    var next_a = actData[acts.length]
                    var actDist = model.af(current_s, current_m, model.params)
                    Observe(actDist, next_a)
                    var next_s = stateData[s_states.length]
                    var next_m = Mode(trial_inference(trial, [m_state, m_states.length], model))
                    var m_dist = model.uf(next_s, current_m, model.params)
                    observe(m_dist, next_m)
                    return iterate(s_states.concat(next_s),
                                   m_states.concat(next_m),
                                   acts.concat(next_a))
                }
            }
            return iterate([stateData[0]], [], [])
        }
        var addFactors = mapN(function(n){
                                   return map(function(y){
                                              return observeTrial(n, y)}, data[n])},
                                              numActors)
        return theory
    })
}
```

The first stage of this program samples a theory from the over-hypothesis, then

samples an actor model from this theory for each actor in the data. This utilizes the *sampleTheory* and *sampleModel* helper functions, which differ depending on which of the four classes of Level 3 inference we are performing (we explain each version in greater detail below). The *observeTrial* function is very similar to the core function for Level 2 inference, except that it is applied to each trial for each actor. Also similar to the Level 2 program, we increase the tractability of this program by applying a MaP estimate of each actor's mental state for each step of each trial (rather than sampling the mental states directly).

With this general form established, we will now explain the details of each form of Level 3 inference.

Single-population parameter inference The easiest Level 3 problem is single-population parameter inference, which corresponds to a case where the observer believes that actors behave according to a particular model structure, and a framework theory specifies the prior probability over the model's parameters. In this case, the over-hypothesis specifies:

- a single model structure M with parameters in θ,
- a family of prior distributions $P(\theta;\gamma)$ over θ parameterized by γ, so that each value of γ specifies a framework theory, and
- a prior distribution $P(\gamma)$ over γ.

The *sampleTheory* function draws a sample of γ from $P(\gamma)$, and the *sampleModel* function draws a sample of θ from the appropriate framework theory $P(\theta;\gamma)$, and plugs this value of θ into the fixed model structure M.

Single-population full-model inference A single-population full-model inference problem corresponds to a case where the observer believes that all actors behave according to the same model structure, but does not know the structure. A framework theory specifies a single model structure out of a set of possible alternatives, and a prior distribution over the parameter space for that structure. In this case, the over-hypothesis specifies:

- a (possibly infinite) set of model structures $\{M_i\}$, and a parameter space θ_i for each model structure,
- a prior distribution $P(M)$ over model structures,
- for each model structure M_i, a family of prior distribution $P(\theta_i; \gamma_i)$ over the parameter space for that structure, parameterized by some γ_i, and
- a prior distribution $P_i(\gamma)$ over each hyper-parameter γ_i

Within this class of inference problem, there are two methods to encode the set of model structures and structure prior. Under the first method, the over-hypothesis specifies a fixed and finite set of possible model structures $M_1, \ldots, M_k$. In this case, the *sampleTheory* function first generates a structure by sampling from a categorical distribution over $M_1, \ldots, M_k$, then samples a hyper-parameter from the appropriate $P_i(\gamma)$, which determines a prior distribution $P(\theta_i; \gamma_i)$ over the appropriate parameter space. The *sampleModel* function then samples a parameter value from $P(\theta_i; \gamma_i)$, and plugs this parameter value into the chosen model structure M_i.

Under the second method, the over-hypothesis specifies a probabilistic generative grammar (PGG) over model structures. A PGG consists of

- An initial "simplest" model structure M_0
- A finite set of possible transformations $T_0, \ldots, T_m$, each of which takes a model structure (and it's corresponding parameter space) and outputs a transformed model structure and parameter space.
- A function $p : \mathbb{M} \to \mathbb{R}^{m+1}$, which specifies, for a given model structure M, the probability that each of the $m + 1$ possible transformations will be applied.

For an over-hypothesis of this form, the *sampleTheory* executes the following recursive procedure:

- Step 0: output the initial model M_0
- Step 1: sample a transformation T_i with probability $p(M_0)_i$, then apply this transformation to M_0 to obtain M_1
- Step n: sample a transformation T_i with probability $P(M_{n-1})_i$, then apply this transformation to M_{n-1}. We designate a single "terminal" transformation T_0, which returns M_{n-1} unchanged. If the terminal transformation is chosen, the recursive procedure halts and returns M_{n-1}. Otherwise, we repeat this step using M_n as the input. We require $p(M_n)_0$ be greater than zero and non-decreasing in n, which guarantees that *sampleTheory* will always halt after a finite number of steps with probability 1.
- Final step: Once the terminal transformation is applied and outputs model structure M (and its corresponding parameter space), *sampleTheory* samples a value of the appropriate hyper-parameter γ_M to determine a prior distribution $P(\theta_M; \gamma_M)$

Once *sampleTheory* is executed and returns a model structure M and parameter prior $P(\theta_M)$, the *sampleModel* function samples a value of θ_M from the parameter prior and plugs it into the model structure M. While we do not present any simulations that use a PGG over-hypothesis, we outline a simple PGG for actor models in appendix B4.

Mixed-population inference Both of the previous forms of Level 3 inference assume that all actors can be explained by a single prior distribution over actor models. Under the mixed-population versions of these problems, the observer assumes that actors may belong to different sub-populations, and the framework theory specifies the number of distinct sub-populations, and for each sub-population, a parameter prior (for mixed-population parameter inference) or a model structure + parameter prior (for mixed-population full-model inference). In this case, the over-hypothesis specifies:

- Either the number of sub-populations, and a prior distribution over sub-populations (i.e. the prior probability that a new actor belongs to a particular sub-population), or a prior distribution over the number of sub-populations. We can use a non-parametric prior such as a Dirichlet Process (Rasmussen 2000) to allow the observer to learn the number of sub-populations directly from data.

- An over-hypothesis for each sub-population, which specifies either a fixed structure and parameter hyper-prior, or a structure prior and family of parameter hyper-priors (depending on the type of Level 3 problem)

If the over-hypothesis specifies a fixed number of sub-populations, the *sampleTheory* function samples a model structure and parameter prior for each sub-population. If the over-hypothesis uses a non-parametric prior, the *sampleTheory* function first samples a

number of sub-populations, then samples a model structure and parameter prior for each sub-population. In both cases, the *sampleModel* function first assigns each actor to a sub-population by sampling from the sub-population prior, then samples a model structure from the appropriate structure prior, then samples a parameter value from the appropriate parameter prior.

B4: Example of a PGG for actor models

Here we provide an example of a simple probabilistic generative grammar that allows us to define a framework theory (prior distribution over actor models) over actor models of arbitrary complexity. To keep this example simple, we shall assume that each model only includes G-states, that each G-state is a VAL variable, and the domain of each VAL variable is restricted to binary state-value functions (i.e. $G(s) \in \{0, 1\}$ for each system-state s). The base theory T_0 consists of the default variables s_t, a_t, s_{t+1}, and a single VAL variable G. The base program for a_t is given by The helper function

---

var $actSelect$ =function($state, G, \gamma, \theta$){
    var $possibleActs = states\_toActs(state)$
    var $stateVals = map$(function(x){return $expUtil(G, state, x, \gamma)$}, $possibleActs$)
    var $actProbs = Transform(stateVals, \theta)$
    return $Categorical(vals$: $possibleActs$, $probs$: $actProbs)$
}

$expUtil(G, state, x, \gamma)$ computes the expected/possible/maximum[76]future utility of an actor with goal G (a binary value function over states) and discount rate γ taking action x from state *state*. The second function $Transform$ converts a vector of utility values (one corresponding to each possible move) into a normalized vector of action

[76]Depending on the form of the observer's built-in utility computation function

probabilities. For example, $Transform$ may apply a softMax transformation with concentration parameter θ, or it may linearly re-scale the values into probabilities and then adjust those action probabilities by error parameter θ. The final line returns a distribution over possible actions, with probabilities determined by the estimated utility values.

The PGG starts with the initial theory above, and then applies a series of transformations via stochastic recursion until a (probabilistic) termination criterion is reached. Each non-terminal transformation involves the following steps:

- Choose a VAL variable G in the current theory T_t. The probability of choosing a particular G is inversely dependent on the number of sub-goals currently under G, i.e. variables with fewer current sub-goals are more likely to be chosen than variables with more sub-goals, and variables with no current sub-goals are most likely to be chosen.

- If G has no current sub-goals:

    1. Create a new sub-goal variable G' consisting of two binary VAL variables g_1 and g_2, conjoined through one of the following operators (chosen at random): *and*, *or*, *then*. The two VAL variables, in conjunction with the binary operator, define a new binary VAL variable: $and(g_1, g_2)$ assigns 1 to any state in which g_1 and g_2 are both satisfied; $or(g_1, g_2)$ assigns 1 to any state in which either g_1 or g_2 are satisfied; $then(g_1, g_2)$ assigns 0 to every state *until* g_1 is satisfied, and then assigns 1 to any state in which g_2 is satisfied.

    2. If G had an arrow into a, replace that arrow with an arrow from G into G', and an arrow from G' into a

3. Modify the program for a so that any reference to the value of G is replaced by a reference to the value of G'. I.e. if the program for a originally included the line $expUtil(G, state, x, \gamma)$, it is replaced with $expUtil(G', state, x, \gamma)$.

- If G already has a sub-goal G':

  1. Let $G' = p(g_1, \ldots, g_n)$ denote the sub-goal expression corresponding to G', consisting of n component VAL variables conjoined by a sequence of binary operators $and, or, then$.
  2. Create a new binary VAL variable g, and compose it with $p(g_1, \ldots, g_n)$, with a randomly chosen operator.
  3. Modify the program for a so that any reference to the sub-goal G' now points to the new sub-goal expression

- If $flip(P(T_t))$, return to the first step with T_{t+1}, otherwise halt and return T_{t+1}. $P(T_t)$ is the halting probability, which increases with the complexity of T_t, so that the process will eventually halt with probability 1.

This generative grammar, in conjunction with a set of transition probabilities (i.e.: probability of choosing a particular G, probability of joining two new sub-goals via and, or, or $then$, probability of a particular joining site and operator for adding a new g to an existing sub-goal sequence, and halting probability), defines a prior distribution over hierarchical goal models with an unbounded number of nested levels (though note that any model with an infinite number of levels has probability 0 under this prior). We can therefore use this PGG to learn an appropriately complex planning model for a given actor.

Appendix C: Simulation details for chapter 4

Here we present the simulation specifications for the case study presented in Chapter 4, which replicates (in simulation) the infant habituation experiments in Woodward (1998).

C1: Derivation of schema space

We define our schema space as the set of all directed graphical models over the variables $X = (s_0, a, s_1, g, \beta_a, \beta_g, \beta_{s_1})$ consistent with the following constraints:

1. Acyclicity.

2. No directed arrows into s_0. This encodes the intuition that the initial state of the environment is fixed prior to the start of the trial and cannot be influenced by any feature of the trial itself.

3. Bias parameters β_a, β_g, and β_{s_1} may only have arrows into their respective features (which follows from their definition as bias parameters).

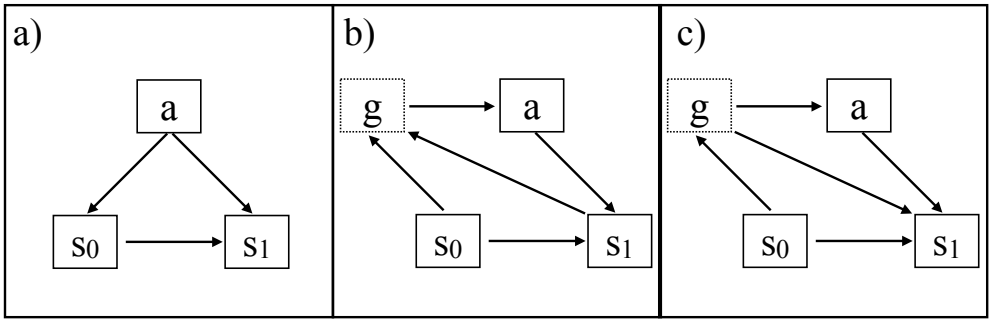
4. If any of a, s_1, or g have no other parent in the structural model, they must have the corresponding bias parameter as a parent. Technically, bias parameters should be present for all features, even if they have another parent. However, these parameters would only be relevant for comparing the behavior of one actor against the behavior of another. As the current case study involves observation of only one actor, we may omit the bias parameters for features with other parents.

5. The induced probability distribution $P(s_0, a, s_1)$ must be consistent with the transition distribution $P(s_1|s_0, a)$. This parametric assumption encodes infants'

knowledge of object physics and the physical principles of reaching (see Leslie 1984).

6. If a model includes the goal variable g, g must be sufficiently strongly correlated with s_1. This parametric constraint encodes the knowledge that goals track outcomes (for infants who interpret the action in terms of a goal-driven agent).

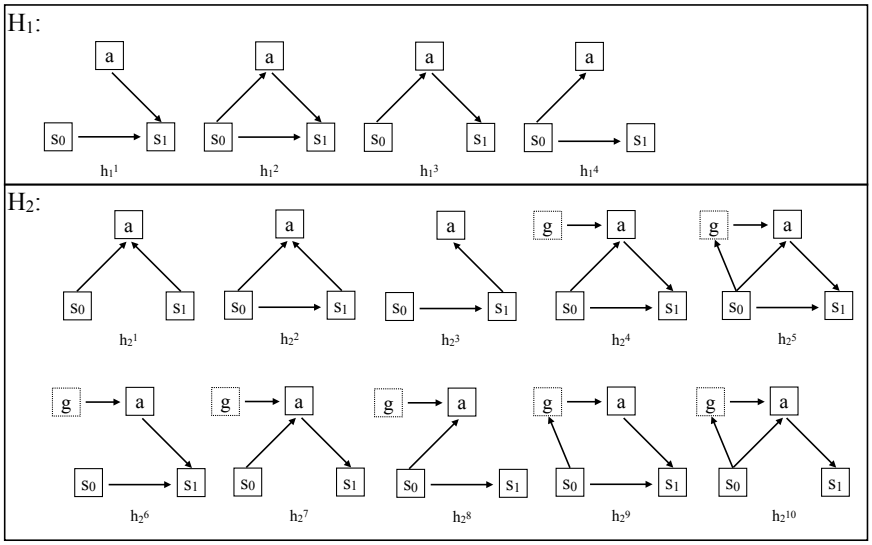
Figure C.1.1 illustrates three models which fail to meet these criteria for different reasons (and are therefore omitted from our simulations). Figure C.1.2 illustrates the 14 models which do meet these criteria and constitute the basis for our simulations. We omit the bias variables from these figures in order to save space.



C.1.1: Three examples of structural models which are not consistent with our constraints. Model a) violates constraint 2, as it contains an arrow from a into s_0. Model b) violates constraint 1, as it contains a cycle. Model c) violates the parametric constraint 5. In particular, this model allows the probability of s_1 to vary even when the values of s_0 and a are fixed, which violates the requirement that the joint distribution over (s_0, a, s_1) be consistent with the transition distribution $P(s_1|a, s_0)$.

C2: Simulation specifiations

All simulations were coded in WebPPL, a probabilistic programming language for generative models (Goodman & Stuhlmuller 2018). For each model M, we compute the



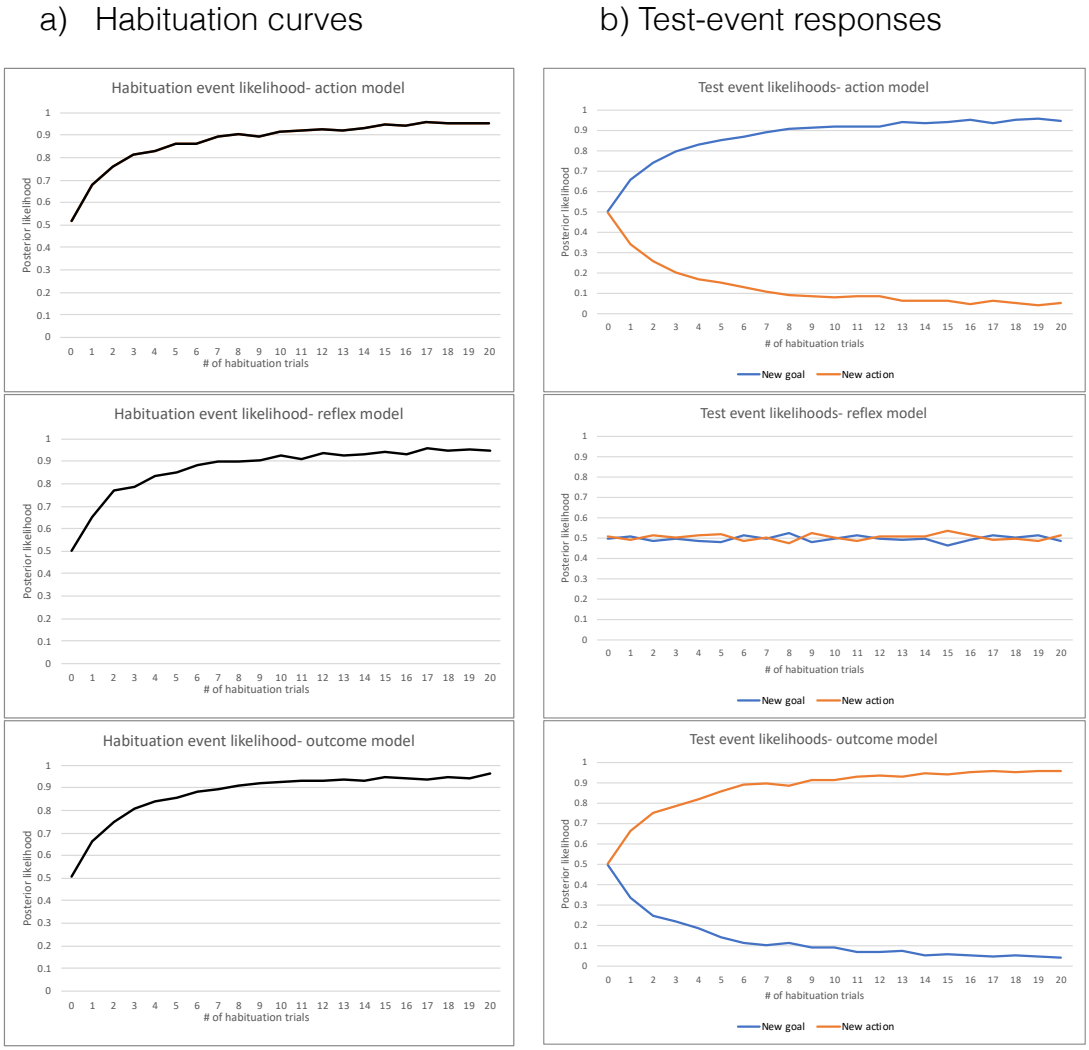
C.1.2: The 14 structural models consistent with our list of constraints

posterior likelihood of the habituation event s, given the observer's prior beliefs (i.e. model M and prior distributions) and n observations of the habituation event; we denote this likelihood $P(s|S_n, M)$. We equate the increase in posterior likelihood with the observer's familiarization to the habituation stimulus. For these simulations, all parameters are drawn from uniform priors. In general, computing the posterior likelihood exactly may be intractable, so we approximate the posterior using a Markov Chain Monte Carlo (MCMC) sampling method, which generates a set of 10000 samples that approximates the true posterior. This step is solely to improve the tractability of the simulations and does not reflect an assumption regarding the observer's cognitive processes. We perform these computations for $n = 0, 1, ..., 20$. In addition to the habituation event likelihood, we compute, for each n, the posterior likelihood of each test event t_1 ("new-goal") and t_2 ("new-action") under the distribution $P(t|S_n, M)$. The ratio of these likelihoods reflects whether the observer shows a preference for t_1 or t_2 (or

neither). In particular, we assume that an observer who attends longer to test event t_1 does so because t_1 is significantly more unexpected than t_2. Thus, we report that the observer "prefers" t_1 if and only if the posterior ratio $P(t_2|S_n, M)/P(t_1|S_n, M)$ is sufficiently larger than 1 (we use a threshold of 1.5 for our results table).

Simulated habituation and test curves

To better illustrate the outputs of each simulation, figure C.3.1 shows habituation and test-event response curves for the action, reflex, and outcome models (h_1^1, h_1^2, and h_2^1 in appendix C1).



C.3.1: simulated habituation and test-event response curves for action, reflex, and outcome models. Panel a) illustrates the steady increase in posterior likelihood, corresponding to the observer's increasing familiarity with the habituation event. This occurs across all models. Panel b) illustrates the posterior likelihood of both test events given n observations of the habituation event. If the observer starts with an action model, the posterior likelihood of "new-action" drops significantly as habituation proceeds, while the posterior likelihood of "new-goal" increases significantly. Thus, as n increases, an action-model observer develops a strong preference for "new-action." On the bottom of b), the response graph illustrates that an outcome-observer would instead develop a strong preference for "new-goal," while the middle panel demonstrates that a reflex-observer would develop no preference for either event (i.e both events are equally unexpected).