# Predictive Representations of State and Knowledge

November 23, 2011

**Abstract**

This paper shows how a single mechanism, called a "forecast," allows knowledge to be constructed in multiple layers, where each layer is built from existing features and knowledge. This predictive representation captures high-level, abstract knowledge exclusively in terms of its most fundamental constituents: low-level senses and actions.

It is generally accepted among those studying general intelligence and developmental learning that high-level knowledge must eventually be connected to low-level sensory and motor data. Though many powerful candidates have been proposed, the search is still on for a robust mechanism that can successfully bridge this gap over a large number of different levels. We believe that forecasts may achieve this long-sought goal. This paper provides a concrete illustration of how an artificial agent can connect its abstract knowledge to its raw sensorimotor experiences, beginning with its sensorimotor stream and representing its knowledge as a set of layered predictions (forecasts) about the consequences of its actions. This illustration shows twelve separate layers, the lowest consisting of raw pixels, touch and force sensors, and a small number of actions; higher layers increase in abstraction, eventually resulting in abstract concepts corresponding roughly to doorways, walls, and rooms.

We further suggest the possibility that this general mechanism may allow the representation of a broad spectrum of everyday human knowledge.

## 1 Motivation and Overview

One of the great problems in AI and Cognitive Science is that of connecting a representation to what it represents. Despite decades of research developing ontologies and datasets for encoding information that most children take for granted, no representation has yet emerged capable of capturing the richness of a child's understanding. As humans, we draw on vast resources of intuition to justify our judgements. We know that bananas are yellow, and that pillows are soft, that a glass can shatter when it is dropped on a hard surface, and that most people like the taste of ice cream. We know these things without having been told them explicitly; we have learned them through our everyday experiences.[1] Yet these facts are represented in us in an accessible way: we can assess their validity quickly, and we can use them for deeper reasoning. Such a powerful representation has eluded the best efforts of AI researchers, but perhaps we are now finally beginning to make progress towards reproducing this power.

### 1.1 Why a New Knowledge Representation?

Reasoning systems in AI rely on information encoded by humans and represented by humans in forms that humans think are useful for reasoning. Adult humans, especially adult human AI researchers, tend to consider *knowledge* to consist of symbols bound together by a network of relationships, while *reasoning* is the application of rules of inference to these symbols and relationships. Yet the symbols themselves tend

[1] The term *everyday experience* refers throughout this paper to low-level interaction with the world, and specifically to the low-level interactive data stream: raw sensory input and motor output.

not to be related to the universal basis of all human knowledge: everyday sensorimotor interaction with the world.

The CYC project—the world's largest relational knowledge base—currently solicits help online to answer queries that cannot be resolved within the knowledge base itself. The website visitor is asked to confirm, if possible, whether a given statement is true. In some cases, the help needed is simply factual, such as:

> Bauxite ore is a natural resource of Greece.
> Aramaic is spoken in Syria.
> Frank Nelson appeared on "I Love Lucy".

In these cases, CYC lacks some specific bits of world knowledge that the visitor is asked to supply. But in many other cases, a different kind of confirmation is requested:

> Most balloons are taller than most cages.
> Most canes are heavier than most bed pillows.
> Most hominids are heavier than most lions.

So what is it that CYC is missing in these cases? Does it require simply more knowledge about pillows, balloons, and lions, or a better definition of heaviness or height? What does the human have that the knowledge-based reasoning system does not? Though perhaps hard to define exactly, there is an obvious intuitive answer here: what CYC lacks is experience with the real world.

Humans are indispensable to CYC's interaction with the world. Humans are not just necessary for encoding CYC's knowledge, but also for interpreting its symbols. This human ability that knowledge bases lack is, perhaps, in some sense, the real story of intelligence. We know approximately how heavy canes are, as well as hominids, lions, and bed pillows—despite not having placed each of them on a scale or having read anywhere what they should weigh. We can imagine interacting with these objects, and this allows us to answer an unlimited number of questions about their properties and relationships. We can consider how heavy, or how soft, or how comfortable, a pillow is by imagining what happens when we manipulate it. We can guess how heavy a lion might be just by imagining the act of trying to lift one up.

Among those in AI and Cognitive Science who study general intelligence and developmental learning, it is generally appreciated that knowledge must eventually be connected with sensorimotor experience, yet the task of making the connection has seemed too immense to bridge. We have lacked the appropriate toolset, language, and framework to make this connection, though attempts have been made for many decades to find them.

Piaget [**?**] proposed that children *construct* their view of reality as they grow. This view, now called *constructivism*, represents a very large and vibrant area of cognitive psychology, and our work can be seen as a step on the constructivist path, or more precisely, as a possible piece of the constructivist puzzle. Piaget and many following in his footsteps have sought a single mechanism or principle to bridge the gap from low-level sensorimotor interaction to abstract thinking. Piaget used the *schema* of Kant to describe the construction of intelligence across multiple developmental stages, but this was not a computational model. Others inspired by Piaget have tried to build computational models based on them, notably Drescher [4], whose implementation of Piaget's schemas within a simulated agent constructed a rudimentary understanding of object permanence, starting from primitive senses and actions. In the 1970's, Cunningham [3] attempted to combine the relatively high-level schemata of Piaget with the low-level "cell assemblies" of Hebb [6] (a more computationally explicit constructivist proposal) to produce a hybrid model of intelligence. Cunningham explicitly assumed and sought "a small basic set of data structures" from which "all the complex structures and operations eventually recognized as being intelligent" could develop. The JCM system of Becker [2] and (though less explicitly) the classifier system of Holland [7] were also originally conceived with similar constructivist goals in mind.

In the 1980's the neural-network resurgence returned the sub-symbolic representational perspective to the forefront of AI, and at roughly the same time, reinforcement learning also experienced renewed interest, as researchers strove to increase the autonomy of artificial agents by allowing them to learn about uninterpreted sensorimotor signals from positive and negative rewards. In addition, handcrafted agents that learned about the world in explicit stages of development [11, 12, 14] helped to sway the community toward an agent-centric viewpoint. Each of these in its way was attacking some part of the constructivist problem, seeking agents that autonomously build up an understanding of the world.

"Continual learning" and "continual development" were the terms first applied to open-ended, hierarchical, constructivist learning set within a modern connectionist and reinforcement-learning framework [17, 18, 19, 20].[2] The resulting agent, CHILD [19], was designed explicitly for continual, hierarchical, incremental learning and development; it started from raw sensorimotor interaction and autonomously built up context-sensitive skills. This and other work from the 1990's on learning to learn [22] led eventually toward theories of optimality in reinforcement-learning agents that begin from sensorimotor interaction alone [8]. Current trends in AI also include methods for agents to learn probabilistic models of their environment through interaction [?], which is often combined with reinforcement learning methods [?]. In addition, at least one annual conference (the relatively recent International Conference on Developmental Learning) is devoted specifically to the problem of modeling cognitive development. And at least one large EU-funded project is mandated to contribute toward constructivist goals [1]. Other cousins of the current paper include recent work on predictive state representations (PSRs) [13], TD Networks [30, 15, 29] and, most directly, Horde [26]. (More will be said about these later). Thus, the constructivist movement has a long history in AI and Cognitive Science, and though the goal of the constructivist program is tremendously ambitious, much progress has already been made.

Yet despite many advances, one goal has remained elusive: finding a single toolset with which new skills and knowledge can be formed from existing skills and knowledge through unlimited levels of meaningful abstractions. While Drescher's schema mechanism succeeded at creating a few clear levels of abstraction, it was less clear that the mechanism was general enough to span large numbers of levels. In contrast, CHILD was able to span an arbitrary number of levels, but it was less clear that these levels were general enough to represent abstract knowledge.

This paper proposes that prediction, and specifically a mechanism we call "forecasts," may provide a representation that allows the constructivist program to proceed, enabling abstract knowledge to be constructed in multiple layers, where each layer is built from existing features and knowledge. Forecasts help organize the world in the way a baby or any beginning creature might, building knowledge out of questions of the form: if I behave in a certain way, what will I perceive? For example, if I tried to lift a lion, how heavy would it feel?

This paper attempts to explain how forecasts work, how they can be used to encode knowledge predictively, how such knowledge is similar to our own, and how the agent can use this knowledge to help it represent its state and choose its actions. The largest contribution of this paper is a demonstration of how forecasts can be built up into increasing layers of abstract knowledge beginning from an agent's raw sensorimotor stream. The agent begins with extremely limited sensors and actions yet is eventually able to build an awareness of its surroundings, understanding where it is within its world, distinguished by rooms of various sizes, doorways, walls and windows.

Our overall goal here is to provide an insight into how knowledge can be built from forecasts about sensorimotor interaction. We hope that it conveys sufficient detail into the process that the community can begin to fill in the gaps that we inevitably leave open. In an article such as this, we cannot present everything that might be possible, instead we hope only to inspire, to give a glimpse of the broad tapestry of knowledge that can be meaningfully represented by prediction.

## 1.2 Predictions as knowledge

How can predictions about the future capture what we mean by knowledge? When we say that a banana is yellow, that a glass will shatter if it falls against a hard surface, or that most people like the taste of ice cream, what are we saying? In each case, the sentence can be turned into a prediction about a possible

[2] The term "continual learning" will be used throughout this paper to refer to the full learning process of the constructivist agent, as it concretely and succinctly captures the following critical attributes: continual and unlimited growth through interactive, autonomous, isolaminar, online, and incremental learning from positive and negative rewards. (See Ring (1994) or Ring (1997) for a precise description and detailed discussion of this term.)

future; each sentence puts into words what we believe we would perceive in a certain future scenario. Thus, we are not making an outright prediction about *the* future—what *will* happen, but are instead making a statement about a set of *possible* futures—what *could* happen.

What distinguishes the set of possible futures from the actual future is our own implied involvement in the prediction, our own interactivity. If we determine an object to be a banana, there is a good chance that we can also perceive it as yellow. If we release (something we've determined to be) a glass and let it fall towards (something we've determined to be) a hard surface, there is a good chance that we will perceive a loud noise and a sudden acceleration of many small things where there had previously been a single thing (that we had determined to be a glass). If we were to give (something we had determined to be) ice cream to (something we had determined to be) a person, then there is a good chance we would be able to perceive an improvement in mood in the recipient (a very complex set of predictions).

Of course, all of these examples entail some rather sophisticated sub-components of their own. Yet we conjecture that these sub-components can also be captured in terms of predictions, and that there will be no infinite circularity; instead, all statements of knowledge are eventually reducible to predictions about raw experience, i.e., to predictions about interactions between our senses and actions.

Therefore, for an agent to have knowledge, it must understand something ultimately about its sensorimotor stream. But if *all* knowledge reaches all the way down to the sensorimotor level, then how can it come about in the first place? How can it be built up?

## 1.3 Building from the Bottom Up

A baby comes into the world knowing very little but having the latent ability to know very much. It has the nascent capacity to represent all the things the baby will come to know during its lifetime, from how its body and environment work, to the objects it will interact with in its environment, to the facts and skills it will collect over the course of its life.

As a child grows, it learns continually, building each stage upon the previous one, each more complex skill and piece of knowledge formed from those that preceded it. But how? We believe the answer can be found in the way a baby, like any beginning creature, must organize its experience to better understand its world.

William James famously wrote that "the baby, assailed by eyes, ears, nose, skin, and entrails at once, feels it all as one great blooming, buzzing confusion" [10]. A baby, initially lost in its confusion, begins to notice regularities, certain relationships between its actions and its perceptions. It cries and it gets picked up; it drinks and its stomach feels full; it moves its eyes to the left, and what was before on the right side of its retinas moves to the centers; it puts its hands together and feels sensations that reliably correspond to its movements. As these experiences are repeated, they become more predictable. The baby comes to expect the resulting observations given the actions it produces.

To understand its world, the baby begins to organize and refine these expectations because it is useful and important to predict what the outcomes of its actions will be, how things will change if it chooses one action over another. Early on, regularities are frequent and coarse, because the range of possible actions is limited. As the baby ages and learns, its actions become more complex and its observations and predictions become more refined.

This paper suggests that a good way for a beginning creature to understand its world is simply by making predictions about the consequences of its actions. Furthermore, this predictive way of representing the world might continue to be useful throughout the various stages of growth, remaining as valuable in the adult's world as in the baby's.

## 1.4 Predicting with Forecasts

Prediction, especially predictive models, are common in AI, but these typically make predictions that are much too precise. Predictive models in robotics and reinforcement learning nearly universally make single-step predictions; i.e., they predict what the agent's next observation will be if the agent takes a specific action in a specific state. Similarly "predictive representations" have been in the literature now for a decade, but (nearly) all previous work has dealt exclusively with single-step predictions.[3] In principle, single-step predictions can be quite useful and can provide the foundation for making more complex predictions. But they are not appropriate for answering the kind of predictive questions we tend to need answered.

In normal life, we need to predict many different kinds of events and quantities within a rather non-specific time frame. We need to know things like whether the door will open if I turn the handle, or what the chance of crashing into another car is if I pull out onto the street right now, or how likely it is that the ball will go into the hoop if I throw it from over here, or how full my stomach will be if I eat the entire piece of cake. In none of these cases, and in very few cases throughout life, is it critical to know the precise time step at which anything will occur, or to know anything about what will happen exactly one time step from now. In general, prediction at fine temporal granularity is something that is technically convenient rather than particularly useful to AI.

In common parlance, the word "prediction," is very broad and can mean many things. A good thesaurus lists a dozen or more synonyms, each slightly different. Most predictive mechanisms in AI refer to just one very specific—and not a very useful—meaning: one-step state predictions. Our approach rests on a very different, but also very special, kind of prediction, and so we distinguish it with a specific word: "forecast," which captures many of the right qualities.

A *forecast* is not a one-step prediction: it does not (generally) estimate the value of a state variable or feature at the next time step. A forecast is instead an estimate of a measurable quantity over the course of the future. Like the traditional use of the word, it refers to statements like "the chance of sunshine on Thursday" or "how much snow will fall in January;" however, we define a forecast to be explicitly conditioned on the agent's activity, so it actually more closely resembles statements such as, "how cold it will it be in the city I am visiting in March," "how wet my clothes will become if I run to the market in this rain storm."

Forecasts are temporally extended, depend on the agent's action and can involve probabilities and cumulative quantities, but are always estimates of future scalar values. As with the weather, there can be an unlimited number of forecasts about an unlimited number of things, but each forecast must adhere to strict formal guidelines that are presented in Section 2.3.

## 1.5 Gaining Experience, Continual Learning, and Isolaminar Construction

As humans develop and build up layers of abstractions, our experience is our only link to our world. Everything we understand, we have learned by interacting with our environment. All our knowledge derives from this small stream of sensorimotor data.

An agent that gains human-like knowledge will need to make use of its experience to build up and refine that knowledge. Thus, as it interacts with the world, it will learn continually—constantly developing, constantly refining what it knows, constantly building on top of what it already has, using what it knows now as the basis for what it learns next. This continual-learning process is critical to the development of knowledge from sensorimotor activity.

We are hardly the first to suggest that an agent's learned models should be refined through experience. A great deal of current AI research is focused on the use of probabilistic methods to refine abstract models of the world using sensorimotor data [**?**]. We are therefore very sympathetic to the goals of this research, but our approach is different in three critical ways: it is based in prediction, it uses forecasts for those predictions, and it relies exclusively on *isolaminar* construction methods. By "isolaminar," we refer to any multi-layer constructive method that uses the same mechanism at all levels of construction. Thus, a brick wall is isolaminar; a suspension bridge is not.

To build knowledge from the bottom up means building at multiple levels, and probably building at many such levels simultaneously. To build at multiple levels requires either inventing new methods at each level to construct that level from the previous one, or using an isolaminar method that will work at every level equally well. Finding such a single method is not easy, because it must be just as suitable for building knowledge from low-level sensorimotor data as from high-level abstractions; it must be equally suited to all

[3] See Section 2.5 for a fuller discussion.

kinds of experience and all kinds of knowledge, from gustatory and kinesthetic knowledge to musical and mathematical knowledge.

In this paper we propose an isolaminar method for representing knowledge predictively, and though we would not yet claim that it successfully meets these ambitious goals, it is certainly directed towards them.

### 1.6 Generalization

Before describing the details of our predictive representation, it is worth considering the properties we want it to have. The representation should :

a) describe knowledge in terms of senses and actions;
b) allow continual learning from experience, including the incremental, isolaminar development of knowledge at different levels of abstraction;
c) afford access to the knowledge through planning and reasoning.

The represented knowledge itself should:

a) result from interaction with the world;
b) be modifiable and extensible with new experience;
c) be useful to the intelligent agent.

We humans are agents in an environment that is unfathomably complex. Our understanding of the universe only captures a minute fraction of the predictable regularities that determine the course of its future. To make choices based on what is likely best for us, we are forced to evaluate the various futures these choices represent, and we must do so using a considerably incomplete representation of our environment. Yet the approximation we have of our world is quite useful to us, even if it is inaccurate and vastly incomplete. The regularities it captures are sufficient for us to make successful and useful predictions.

In other words, our representation generalizes well.

A representation that generalizes well captures useful regularities of the environment in a way that allows accurate prediction of the consequences of actions, despite incomplete information.

It is our contention that this is also a good definition of knowledge.

The next section (Section 2) describes a framework for representing general predictive knowledge, then the following section (Section 3) demonstrates how this framework can be used for the isolaminar construction of various kinds of everyday knowledge. After the demonstration, we describe how state can be represented with predictive knowledge (Section 4), and finally, we consider the possible limitations of predictive representations (Section 5), as a result of which we discover that some surprisingly deep and sophisticated kinds of knowledge can be captured quite naturally through prediction.

# 2 An Isolaminar Framework for General Predictive Knowledge

How can knowledge be represented as predictions? This paper is an attempt to answer that question, and the technical details are shown below, but the answer is—conceptually, at least—quite simple. First, build a forecast function to continually estimate the future value of any quantity of interest. Second, learn behaviors (policies) to maximize or minimize the most interesting forecasts.[4] These two steps can be interlaced continually.[5]

This paper describes General Predictive Knowledge (GPK), a framework for specifying isolaminar predictions and learning the correct values for those predictions. Learning is done using recently introduced off-policy temporal-difference methods; in particular, GPK relies on a version of the GQ algorithm in which forecasts are calculated as a function of states rather than as a function of state-action pairs.[6] GPK learns and fine tunes knowledge by improving forecast functions based on the sensorimotor stream. Furthermore, all of the knowledge that GPK learns is verifiable through the evidence obtainable from the sensorimotor apparatus.

## 2.1 Forecasts

It is convenient to think of forecasts as having two different parts. One part is the specification of what is forecasted; for example: "The probability that the glass will shatter if I let it go," or "How soon I will ski into a tree if I lean to the right." This specification is called the "forecast function." The other part is the quantity estimated by the given forecast function, for example 0.99 or 27, which can be called the "forecast value," the "forecasted quantity," or (most frequently) simply the "forecast."

An agent can have a theoretically unlimited number of forecast functions, each estimating a different quantity. Forecasts are updated with *targets* that reflect the forecasted quantity conditioned on a particular kind of behavior. The choice of target and behavior defines the forecast function, because the forecast itself is learned from experience by a function approximator. Both the target and the forecast are computed from the current state information, and the forecasts are learned by the method of temporal differences [23]. Furthermore, behaviors can be learned that maximize or minimize a forecasted quantity. (More will be said about state information in Section 4.)

Thus, to make a forecast means first specifying the target and then learning to estimate that target. Different mechanisms are involved in these two operations. GPK provides both a language for specifying the target quantity and a mechanism for estimating it.

## 2.2 Forecast Functions

Forecasts fall into three basic kinds: forecasts of single *perceptual events*; cumulative forecasts; and mixed forecasts. An *event* forecast predicts the expected value of a specific, single observation or forecast at the

[4] Though this paper describes both of these steps, it does *not* deal with the very important question implied by them: how can the agent decide which quantities and forecasts are interesting? We consider the discovery of new, interesting, and useful forecasts to be an open question. Furthermore, our purpose in this article is to show how the isolaminar, continual-learning process described above can capture surprisingly rich knowledge, fully connected to the sensorimotor data, and to do this we have chosen the forecasts by hand.

[5] The agent is then free to follow a policy or not depending on the extent to which the event becomes desirable. The agent can also compare courses of action with the policy that achieves the desired event.

[6] It is not an intention of this paper to introduce GPK as a novel learning algorithm, but for many expositional reasons, it is more convenient to discuss forecasts as using a state-based description than the action-based one of GQ.

conclusion of a specific behavior; a cumulative forecast predicts a quantity that accumulates over time while the behavior is ongoing; and a mixed forecast combines these together. An example of an event forecast is, "I will see the patio light come on if I flip one of those switches," or "I will find food if I go to the party tonight." An example of a cumulative forecast is "How long it will take me to find the patio switch," or "How much food I will see at the party tonight." And an example of a mixed forecast is, "How many lights I will have turned on when I finally stop flipping the switches," or "How much I will eat at the party tonight" (knowing that I may stop eating either because the party ends or because I am too stuffed). {It's very hard to find examples that also makes sense in the third case.}

**Event forecasts.** All predictions are useless if they do not in some way constrain or refer to the period of time in question. If I predict that there will be an earthquake in California, the prediction is meaningless without reference to the relevant time frame. When I predict that a person's mood will improve if I give the person ice cream, this improvement should fall within a few minutes' time, while if I say the glass will break if it hits a hard surface, I expect it to break the moment it hits the hard surface, not minutes later. Thus, our representation have somehow include the likely time span of the prediction.

Event forecasts mix together the notions of time and probability. The forecast for an event that is certain to occur immediately is 1. The forecast for an event that will certainly never occur is 0. The forecast for an event that is certain to occur at some distant point in the future (e.g., dying) is a low value, just like events that are very unlikely, but could happen immediately (e.g., being struck by lightning). This mixing of time with probability allows the expression of temporally indefinite predictions. As an event's probability of occurence *or* its temporal nearness increases, so does the forecast for that event. If after taking an action, an event forecast decreases, the action has either delayed the event, made it less likely to occur, or both; for example, shifting my weight to my left leg has reduced my forecast of crashing into a tree; or, tightening my grip on the glass has reduced my forecast of hearing a crashing sound.

Different forecast functions can specify different degrees of temporal precision for the same perceptual event, and thus time can be distinguished from probability by combining forecasts of different temporal precision. (See Sutton et al [26] for a demonstration of this principle.) In addition, event forecasts estimate the value of a perceptual event (a specific observation or forecast) when the agent stops behaving in a particular way; i.e., when it stops following a particular policy. The agent can consider many different policies simultaneously (e.g., skiing slowly or fast), and these policies can all terminate under various conditions. For example, assuming my policy for skiing terminates if I crash into a tree, it might be useful to forecast the likely pain I will feel in my nose if I follow one policy or another until termination. Thus, an event forecast captures a general notion of the likelihood of a future perceptual event without specifying the exact moment or set of exact moments when the event will occur.

**Cumulative Forecasts.** It is often useful to predict how much of some quantity will accumulate during a certain activity; for example, "how much food I will eat at dinner tonight." These cumulative forecasts are particularly useful for measuring time or distance; for example, "How much time will elapse before I see the ball hit ground if I throw it into the air," or "How many steps it will take me to get to the door from here," or "How much time I will have after the light turns yellow to make it through the intersection."

**Mixed Forecasts** {This is a tough one; I really don't use this case, so coming up with reasonable examples is difficult, but I also don't want to pass it off or say that it's a special or rare case. For all I know, it could become the most common kind of forecast.} Mixed forecasts are simply combinations of event and cumulative forecasts. They estimate the total cumulated amount of a quantity that accrues during an activity plus a (same or different) quantity observed at termination of the behavior. This can be useful, for example, when termination of the behavior includes the quantity being accumulated, such as when the agent stops flipping switches because it has finally found the right one, or stops eating because it has eaten enough.

## 2.3 Details: Forecast Functions

The algorithm is described within the framework of a Markov decision process (MDP), which, briefly, consists of a set of states ($s \in \mathcal{S}$), actions ($a \in \mathcal{A}$), observations ($o \in \mathcal{O}$) and rewards ($r \in \mathbb{R}$). At every time step, the agent receives an observation $o_t$ in its current state $s_t$ and takes an action $a_t$ which leads the agent to the next state $s_{t+1}$ and reward $r_{t+1}$.[7] Within this framework, each forecast function consists of two parts, an *option* and a *target*, described next.

The *option* [28] is a 3-tuple $(\pi, I, \beta)$, where $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the *policy* (a distribution over states and actions) that specifies a way of behaving; $I : \mathcal{S} \rightarrow \{0, 1\}$ is the *initiation set*, the states in which the policy can be started (though it may continue in other states outside of the initiation set); $\beta : \mathcal{S} \rightarrow [0, 1]$ is the *termination probability*, the probability of the option terminating in each state. The option describes a possible way for the agent to behave, along with information about where that way of behaving can begin and end. Each forecast function is only meaningful with respect to its option and estimates the value of the target when the option is followed. A good way of thinking about the forecast function is that it poses the question, "if I act in this way, what will the outcome be?" (Thus, forecasts here are often conveniently stated as questions.)[8]

The *target* is a tuple $(c, z)$, defined for every state, where $c : (\mathcal{S} \times \mathcal{A}) \rightarrow \mathbb{R}$ is a *cumulative* value defined for every state-action pair reachable while the option is being followed, and $z : \mathcal{S} \rightarrow \mathbb{R}$ is a *termination* value, defined wherever termination may occur. The forecast is computed as the expected sum of all the cumulative values while the option is being followed, plus the termination value at option termination. Thus, every forecast $f^i$ is a function of the state as specified by the five components of the forecast function:

$$f^i(s) \equiv f^{\pi^i, I^i, \beta^i, c^i, z^i}(s)$$

Given that the following applies only to forecast function $i$, the superscript $i$ has been dropped for clarity. For the current state $s_0$, and for termination at future time step $k$,

$$f_k(s) = \mathbb{E}\left[c_1 + c_2 + \ldots + c_{k-1} + z_k \mid \pi, \beta, s = s_0, \text{termination at } k\right] \tag{1}$$

However, $k$ is unknown and can be any non-negative integer; therefore

$$\begin{aligned} f(s) &= \mathbb{E}[f_k(s) \mid \beta] \\ &= \sum_{k=0}^{\infty} P(\text{termination at } k) f_k(s) \\ &= \sum_{k=0}^{\infty} \sum_{s' \in \mathcal{S}} P_k(s, s') \ \left[\beta(s')z(s') + (1 - \beta(s'))c(s')\right], \end{aligned} \tag{2}$$

where we view the states that the agent visits when following the policy $\pi$ as a Markov chain with a special absorbing state, $s^\circ$, in which $c = z = 0$ and to which the MDP can transition with probability $\beta(s')$ from every state $s'$ reachable from $s \in I$. $P_k(s, s')$ is the probability that the agent will be in $s'$ exactly $k$ steps after visiting $s$ (while following $\pi$); and $c(s)$ is the average reward received by taking a policy action in $s$. It

[7] This paper does not deal extensively with the reward, but it is included for completeness. Also, it is not critical that the agent has access to the full state information (even though the existence of such underlying states is presumed by the theoretical framework), because useful partitions of the state space can be drawn without it.

[8] In fact, forecasts are called "questions," in the Horde architecture [26].

is assumed that all options will eventually terminate, thus avoiding any infinite sums.[9]

$$
\begin{aligned}
f(s) &= \sum_{s' \in \mathcal{S}} \sum_{k=0}^{\infty} P_k(s,s') \ \left[\beta(s')z(s') + (1-\beta(s'))c(s')\right] \\
&= \sum_{s' \in \mathcal{S}} \left[ P_0(s,s') \ \left(\beta(s')z(s') + (1-\beta(s'))c(s')\right) + \sum_{k=1}^{\infty} P_k(s,s') \ \left(\beta(s')z(s') + (1-\beta(s'))c(s')\right) \right] \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s' \in \mathcal{S}} \sum_{k=1}^{\infty} P_k(s,s') \ \left[\beta(s')z(s') + (1-\beta(s'))c(s')\right] \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s' \in \mathcal{S}} \sum_{k=1}^{\infty} \sum_{s'' \in \mathcal{S}} P_1(s,s'')P_{k-1}(s'',s') \ \left[\beta(s')z(s') + (1-\beta(s'))c(s')\right] \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s' \in \mathcal{S}} \sum_{s'' \in \mathcal{S}} P_1(s,s'') \sum_{k=0}^{\infty} P_k(s'',s') \ \left[\beta(s')z(s') + (1-\beta(s'))c(s')\right] \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s'' \in \mathcal{S}} P_1(s,s'') \sum_{s' \in \mathcal{S}} \sum_{k=0}^{\infty} P_k(s'',s') \ \left[\beta(s')z(s') + (1-\beta(s'))c(s')\right] \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s'' \in \mathcal{S}} P_1(s,s'')f(s'') \\
&= \beta(s)z(s) + (1-\beta(s))c(s) + \sum_{s' \in \mathcal{S}} (1-\beta(s)) \sum_{a} \pi(s,a)Pr(s,a,s')f(s') && (3) \\
&= \beta(s)z(s) + (1-\beta(s)) \left(c(s) + \mathbb{E}[f(s' \mid \pi, \beta, s = s_t, s' = s_{t+1})]\right). && (4)
\end{aligned}
$$

This holds for any state $s'$ reachable from any $s \in I$ before termination of $\pi$. While the agent is following $\pi$, this recursive definition allows a temporal difference update (just as with GQ), because the states are experienced with those probabilities given in Equation 3. However, when the agent is following a different policy, then corrections are required to compensate for the differing distributions. This is what GQ (and therefore GPK) does.

Thus a forecast is a prediction about the value of $z$ at option termination plus the summed values of $c$ while the option is running. For many predictions, it is useful for $z$ to be zero everywhere. For example, the following prediction is a useful description of the distance to the nearest wall.

Option: "Walk to nearest wall"

$I = 1$ if there is a wall nearby, 0 otherwise.
$\pi$ = turn to face nearest wall, or, if already facing the nearest wall, take a step forward.
$\beta = 1$ when at wall, 0 otherwise.

Target:

$c = 1, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
$z = 0, \forall s \in \mathcal{S}$

This forecasts the total number of elapsed time steps if I walk to the nearest doorway and then stop.

[9]One way to implement this in practical applications—though not in this paper—would be to constrain $\beta$ to (0,1].

For other predictions, it is useful for $c$ to be zero everywhere, such as for the forecast that predicts whether my coffee is hot. In this instance it might be useful to reuse an option that already exists for another purpose, namely, the option for drinking coffee.

Option: "Drink coffee"

$I = 1$ if I can detect my cup of coffee nearby, 0 otherwise
$\pi =$ grasp coffee cup, raise it to mouth, pour into mouth, swallow
$\beta = 1$ after swallowing or if my heat sensor in tongue is too high, 0 otherwise

Target:

$c = 0, \forall s \in \mathcal{S}$
$z =$ heat sensor reading from tongue

In other words, the forecast function asks: if I drink the coffee, how warm will my tongue sense it to be?

Thus, each forecast function is described by a 5-tuple $\{\pi, I, \beta, c, z\}$. This formulation provides a broad language that allows specification of a very wide range of layered forecast functions. As we will see in the demonstration section below (Section 3), complex knowledge can be encoded as a set of layered forecast functions. But it is the pattern of forecasts themselves (the scalar values) that encode what the agent perceives at any moment. These forecasts are simply estimates, based on the agent's current state information; they are mappings $\mathcal{S} \rightarrow \mathbb{R}$, from states to scalar values.

## 2.4 Algorithm Details: Forecast Estimation

The forecast $\hat{f}(s)$ is an estimate of the true expected value for the forecast function $f(s)$ as defined by Equation 2. In general, the true value is never known exactly; but it can be estimated from experience. Typically, the estimate is computed based on the information the agent has available to it: its full estimate of its current state, which includes (but is not limited to) all of its sensory input. Much more will be said about the agent's state representation in Section 4, but for now it is important to understand how forecasts are estimated and learned.

The predictive agent represents its state as a vector, $\mathbf{s}_t$, a complete set of values encoding all the information the agent currently has about its environment at the current time step, including its lowest-level sensory information (which might also include proprioceptive action information) and its current forecasts. Each forecast, $i$, is computed as a parameterized mapping from this state information to the forecast target. Learning can be done by online gradient-descent, modifying the parameters to reduce the prediction error.[10] In the following sections we consider each forecast estimator $\hat{f}^i$ to be a multi-layer perceptron (MLP); however, as long as the parameters of $\hat{f}^i$ can be estimated from experience in an online, off-policy fashion, $\hat{f}^i$ can be implemented by any function approximator. The target, $\xi_t^i$ is formed according to the description of the forecast function of Equation 4, and the output of the estimator is compared to the target to produce the TD-error, $\delta$:

$$\xi_t^i = \begin{cases} z_t^i & \text{if } \pi^i \text{ terminates at } t, \\ c_{t+1}^i + \hat{f}^i(\mathbf{s}_{t+1}) & \text{otherwise;} \end{cases}$$
$$\delta_t^i = \xi_t^i - \hat{f}^i(\mathbf{s}_t).$$

The function approximator $\hat{f}^i$ is then modified to reduce $\delta^i$.

It is important to any practical learning apparatus that experience be used as efficiently as possible, and therefore it is useful for all forecasts to be updated from all usable data. Thus, the forecasts for "the number of steps I will need to get to the kitchen," and "the number of steps I will need to get to the dining room" may share a common path. If the agent is currently following that path, then its forecast about both

[10] It is therefore critical that the predictions correspond to measurable quantities, because measurements are required to supply the targets needed to calculate error, update the parameters, and improve future estimates.

options should be improved using the same data. Assuming a very large number of forecasts, it is essential for each to learn from all applicable experience. In other words, learning should be *off policy* when possible. A primary aim of GQ (and GPK) is to learn forecasts through off-policy updates, which is non-trivial, since, until recently, all online, off-policy TD learning methods for function approximators were known to diverge. Recent research [**?**] has overcome this obstacle, and learning methods are now known for the off-policy case.

The MLP is assumed to consist of at least one hidden layer and can thus learn arbitrary mappings from input to target. The forecast may be any arbitrary function over any of the information contained in the agent's state vector, including sensory information and other forecasts. For example, the agent's visual recognition of a doorway from a distance may be based on retinal input as well as other estimated values that the agent has computed, such as the agent's forecasts regarding how many steps it needs to get to the nearest wall or its forecasts regarding other objects in its vicinity. When the agent observes the doorway from different positions, each of the corresponding state vectors are mapped to predictions about the ability of the agent to interact with the doorway, and since the agent's ability to interact does not change much in general from moment to moment, the agent learns to map these state vectors including the different visual inputs to very similar forecasts.

If the agent is dropped suddenly into a new environment where its initial information about a nearby doorway is exclusively visual, the agent will nevertheless produce forecasts about its current situation, including about the doorway (i.e., what the agent's interaction with the doorway would probably be like). Indeed, predictions about objects and properties of the world can generally be made in this way, based on observation, rather than by direct memorization of recent experience alone. As information becomes available to the agent, the forecasts acquire values consistent with each other and with the data seen so far, reinforcing themselves through a process of *perceptual lock in.* (See Appendix A.1 for further details.) From a single glance the agent can reconstruct an entire web of related forecasts that can be verified by interaction. We refer to this process as "perceiving" and we refer to the full web of related forecasts as a "perception." From this perception, the agent recognizes an entire *pattern of interaction* that suddenly becomes available to it. Thus, forecast functions define the agent's expectations about how it can interact with the things in its environment, but the perception of specific things corresponds to the constantly changing forecasts, which ebb and flow in the mind of the agent from moment to moment.

Once a forecast becomes reliable, it can be used as the basis for building new forecast functions. For example, if the agent has forecasts that describe how it can interact with a doorway, it can build other forecasts such as "how long it will take me to reach a doorway from here," or "if I try to go through the doorway I will bump my nose." Notice that predictions such as these, while expressed here from the subjective view of the agent, might tempt us to re-interpreted them (peering into the agent's data structures) as corresponding to objective descriptions such as "robot's distance to the doorway," or "the door is closed," but there is no guarantee that many or even *any* of an agent's knowledge will so readily correspond.[11]

Each of the agent's predictions is learnable and verifiable through its own sensorimotor experience. It never relies on labelled data or human validation of its knowledge. Like the rest of us, the agent has only its experiences to help it make sense of its data stream, and ultimately, verification through interaction is the only recourse the agent has to ascertain the extent to which its predictions are accurate.

## 2.5 The Power of Forecasts

Forecasts are different from the most common existing predictive representations—Observable Operator Models (OOMs), Predictive State Representations (PSRs), and TD-Networks (TD-Nets)—in two significant ways. First, OOMs [9], PSRs [13], and TD-Nets [30] are based on single-step predictions and thus do not capture the temporally indeterminate nature of forecasts. Second, these methods condition predictions on chains of low-level, primitive actions, whereas forecasts are conditioned on options, which can describe complex, extended ways of behaving.

[11] One suggestion of this paper is that it is perhaps the confusion between our subjective descriptions (which are personal yet verifiable from our sensorimotor stream) and our objective descriptions (which promote consensual and social agreement, but are not verifiable from our sensorimotor stream) that has lead to some seemingly insoluble problems in AI, particularly symbolic AI, when objective definitions are sought for ultimately subjective phenomena.

Of course, with a highly accurate model of single-step transition probabilities, it is theoretically possible to estimate long-term probabilities by tracing the paths of all possible futures for every possible sequence of primitive action. But this brute-force approach is computationally infeasible, becoming more and more so as the temporal resolution of the model increases. For example, when juggling three crystal glasses over a marble floor, it may be useful to the agent, as it plans out its moves, to know the probability that a glass will to crash to the floor and break. But there may be an enormous number of possible action sequences that will cause one or more glasses to shatter, and calculating all of these is probably not tractable, especially at high temporal resolution. Going to the airport, eating dinner, driving home, writing a paper—these are all activities that we can make forecasts about, but which would be completely infeasible if modeled at fine temporal resolution, where every possible step of every possible eventuality would be considered.

Generally, the solution to this problem is to choose a level of resolution manually (and carefully) to meet the requirements of the system and the task to be performed. If it is a system for playing chess, for example, the detailed movement of the chess pieces and the players' hands are irrelevant and do not need to be modeled. A juggling system might only need to know the location and velocity of the balls and hands, ignoring everything else. Planning a trip home or to the airport could be reduced to a few key steps for a nice powerpoint presentation. In these cases the designer has the luxury of working within a small special-purpose domain. Probabilistic approaches to experiential AI often rely on this approach for constraining the learning problem [**?**]. But in the general, isolaminar case, the system needs to represent many layers of abstraction at multiple, non-interfering time scales.

Furthermore we often want to compare long-term predictions quickly and easily for different behaviors we execute: what if I move my arm a bit this way or that way—will the probability of a crash increase or decrease? To make such comparisons with a one-step model, the probabilities for every interesting event must be calculated anew at every moment, which takes the brute-force approach from merely impractical to purely science fiction.

Yet this kind of prediction is exactly what forecasts allow. Forecasts keep ongoing estimates, updated incrementally, of future quantities and probabilities, and these estimates are conditioned on different ways of behaving. Furthermore, forecast updates and learning updates are computationally tractable, avoiding the explicit modeling of multiple paths based on single-step transitions. The forecast is at the heart of predictive knowledge and is what is most distinctive about the representational approach presented here.

A predecessor to forecasts was described by Rafols [15] and Sutton et al [29]. Their system combined options and TD-Nets to make option-dependent predictions of future values. Like forecasts, this network made long-term rather than one-step predictions and the resulting agent developed a rudimentary understanding of its surroundings in a "compass world," where it learned to predict and keep track of what it would see if it were to turn and walk to nearby walls, each colored differently but only visible at point-blank range. Forecasts are predictions of a slightly more general nature, and the current article shows, importantly, how these more-general forecasts can be layered to construct isolaminar knowledge. The closest relative to the current paper is the Horde architecture [26], which has demonstrated how GQ can be used effectively to answer arbitrary predictive questions (i.e., to make arbitrary forecasts). Horde is a feasibility study and shows how large numbers of forecasts can operate simultaneously, each predicting the outcome for different options, each learning policies for the predicted target. The current article describes how these forecasts can be layered to produce meaningful knowledge.

## 2.6 Composing Forecast Functions

A novel and particularly powerful property of forecast functions is how they can be combined together, and it is this compositional power that introduces abstraction into the representation. While many standard learning algorithms allow the composition of intermediate outputs (for example, the hidden units of neural networks), little work has been done to allow the composition of *targets*, where new targets are built from learned state variables. Because forecast functions are defined by an option and a target, and because the targets can be built from existing forecasts, new forecast functions can be constructed that predict the values of existing forecasts. For example, the system may be able to forecast the probability of being able to sit down comfortably. This "comfortable sitting" forecast may serve as an input to help in the estimation of other forecasts (this is composition of outputs), but it can also be used as the $c$ or $z$ for specifying a new forecast function. For example, the agent may find it useful to predict when the "comfortable sitting" forecast will be high, because doing so helps it locate itself within its surroundings; it might build another forecast that predicts whether walking through a doorway will result in a high "comfortable sitting" forecast, then this new forecast might provide a clue about how to find to a nice place to plug in its power cord. Existing forecasts can also be used as the termination condition, $\beta$, and the initiation conditions, $I$, such that new options (and therefore forecasts) terminate or begin based on the values of existing forecasts.

The composition of forecast functions is the isolaminar construction method demonstrated in the next section. It is the central method of abstraction, the glue that binds together all the different layers, allowing the new to form on top of the old.

# 3 Demonstration

This section demonstrates how forecasts can be layered to create abstract knowledge. It is the heart of the paper and its most important contribution. Whereas previous constructivist attempts to find an isolaminar toolset have been partially successful and have allowed some limited layering of skills, they could not provide a detailed example that demonstrated large-scale construction, starting from the raw sensorimotor stream and building up to broad, real-world knowledge. The following demonstration attempts to show that forecasts do provide such a toolset and can readily span multiple levels of abstraction to produce layers of increasingly refined, increasingly abstract knowledge.

The demonstration describes twelve individual layers of knowledge in twelve subsections. Each subsection, each layer, consists of a running dialog between two parallel worlds: *the real world* and a *microworld.* In "The Real World," you and I are the agents; we have everyday experiences and everyday knowledge related to those experiences. In this world, we can talk about our perceptions using a common terminology that helps us understand the other's experiences. The "Microworld" mirrors our real world and contains many of the same regularities that we experience, yet is simple enough that it can be described exhaustively within the scope of this article using predictive representations. In the microworld an artificial agent interacts with an artificial world where we are the designers and have perfect knowledge, something not attainable in our own world.

**The Real World.** The "Real World" part of the dialog gives plausible predictive descriptions of our everyday knowledge. Say we want to represent the piece of knowledge, "my keys are in my pocket." This statement specifies a fact about the world and is also verifiable: my keys are in my pocket if I can put my hand in my pocket and find them there. This verification can be represented as something like a forecast: "if I put my hand in my pocket, I will detect my keys." If the probability is high, then I might express this by saying, "my keys are in my pocket." From a predictive viewpoint, the everyday knowledge that I express in this way *is the same as* my prediction that I will feel the keys in my pocket if I search for them there.[12] Then, if I am constantly maintaining my forecasts, and one forecast predicts whether I will detect my keys if I put my hand in my pocket, I can also partially predict *how* I can find my keys by examining that forecast.

This simple example shows the general mechanism for knowledge as prediction that we suggest may underly all knowledge. The following sections provide a broad collection of examples. In every case, the mechanism is the same as that just given.

**Microworld.** The microworld is a small computational world that captures many of the regularities of our own world, including walls, doors, windows, and rooms of various sizes and connectivity. It contains a predictive agent that understands these things in terms of how it interacts with them. The agent's knowledge is similar to our own in many ways, but is shown in a series of tables as explicit GPK forecast functions and options. To make them easier to comprehend, these functions and options are given names, often containing

[12] This is a strong statement. It implies that knowledge is, fundamentally, a set of predictions about our sensorimotor stream. We may utter statements such as "My keys *are* in my pocket," but what we mean, fundamentally, is, "I believe that if I execute a certain set of actions I will make a certain set of observations." This does not prohibit us from speaking as we do and from making statements about the world phrased in the more convenient language of common usage, though ultimately, to the extent the statements have any meaning, they boil down to predictions about our sensorimotor stream.

words like "doorway" and "wall," but these names are only a mnemonic convenience to help us remember them better. The semantics of the knowledge is given exclusively by the forecasts themselves and by the agent's interaction with its environment.[13]

## 3.1 The Foundation: The Sensorimotor Stream

Every agent has a basic set of senses and actions through which it is able to perceive and affect the world around it. The human sensorimotor apparatus is complex and subtle; the agent of the microworld is extremely simple. Yet each are able to use their interface with the world to build up a rich understanding of their surroundings.

**Real world, Layer 0.** In the real world we have an enormous variety of actions and sensations available to us. We use these actions and observations to piece together a picture of how the world works, and that picture is vast compared to the relatively tiny realm encompassing what we can observe at any particularly moment.

**The Microworld, Layer 0.** In the microworld, the agent has much more limited senses and actions than we do, yet it can also piece together a picture of how its world works. The primitive sensory signals form the agent's initial *knowledge units* (KUs). A knowledge unit is any observation (primitive sensory signal) or forecast, or any function of observation and forecasts that can be computed by an arbitrary MLP. In all cases, the values computed by a forecast function defined anywhere in the text may be used as input to a forecast function defined at any later point in the text (including the very next line) and is available at *the same time step.* Thus, if $\mathrm{KU}^i$ is defined in the text before $\mathrm{KU}^j$, than the value of $\mathrm{KU}^i_t$ can serve as an input to $\mathrm{KU}^j_t$.

Table 1 summarizes the primitive KUs, the observations. All KUs are written in bold small capitols (e.g., SEE CHANGE and **SC**). Many KUs have abbreviations that will be used for reference in later, more complex KUs and options. For sensory input, the agent has a fingertip, from which it receives touch signals (TOUCH). It also has a force sensor associated with its touching mechanism (FORCE); so it can push against something and observe how hard it is pushing. The agent has a retina behind a fixed eye that points forward, through which it can see a small part of the world, including its finger. If something moves, the agent can tell this by how much the image on its retina changes (SEE CHANGE or **SC**), using a built-in mechanism that filters out finger movements and noise from the sensor apparatus and the environment.

| Primitive knowledge units (KUs) (observations) | Abbreviation | Range |
|---|---|---|
| TOUCH | | {0,1} |
| FORCE | | (0,1) |
| SEE CHANGE | **SC** | {0,1} |

Table 1: Primitive sensory information is the first step of knowledge development. This is the complete list of sensory signals. All complex knowledge is derived from these primitive units of knowledge (KUs).

Besides primitive observations, the robot has primitive motor commands, summarized in Table 2.[14] Primitive actions and options are written in italics (e.g., *move finger forward*) as are their abbreviations when given (e.g., *ff*).

In some cases it is useful for the agent to predict what will happen if it does nothing, thus there is a *null* option that does exactly that.[15] The agent can move its finger forward and touch things. (The finger automatically retracts when the agent is no longer moving it forward.) Finally, the agent can move through its world as though riding a wheel chair: it can roll forward and backward, and it can rotate a small

[13] This is in contrast to traditional symbolic knowledge bases, such as CYC, whose semantic interpretation relies on the human-readable names attached to the entries.

[14] There are two kinds of tables in this section: those for KUs (as described above) and those for options.

[15] Though for our microworld it is convenient to list the null option as a primitive action, it has a fairly general use in any world, and in the general case should be thought of as a policy that has a low probability of causing sensorial change.

amount (1°) left or right from its current position. Its shape is essentially cylindrical, having a round base with its eye and finger sitting over that base. The *forward* direction is the direction that its eye and finger point, and all angles are relative to forward.

| Primitive Actions | Abbrev |
|---|---|
| *null* | $\varnothing$ |
| *move finger forward* | *ff* |
| *roll forward* | $\uparrow$ |
| *roll backward* | $\downarrow$ |
| *rotate slightly right* (1°) | $\curvearrowright$ |
| *rotate slightly left* (1°) | $\curvearrowleft$ |

Table 2: All complex actions are derived from these primitive motor commands. Each of these primitive options can also be viewed as an option, where $I = \mathcal{S}$ and $\beta = 1, \forall s \in \mathcal{S}$.

### 3.2 Learning and Development

As stated above, this paper describes the knowledge that can be represented and learned using GPK; it does not address the discovery of new knowledge. Nevertheless, a discovery mechanism is envisioned whereby, through a process of continual learning, new knowledge and skills are built from those that already exist. [19] To highlight this process, the following exposition also builds upwards from simpler to more complex knowledge broken into layers, where each layer relies on the layers presented earlier in the process. Thus, we start with pieces of knowledge that would likely be learned early.

**Real world, Layer 1.** In the real world, we probably first learn something about the relationships between our own sensors and effectors before we can begin to learn about the world around us. We come to understand that when we move our hands in certain ways, we occasionally receive unusual sensory input from our fingers; that when we move our eyes, the image on our retinas changes. In short, we learn that through our actions we can sometimes control the sensations we receive.

**The Microworld, Layer 1.** The agent's most fundamental knowledge describes the most predictable relationships: the effects of its primitive actions on its primitive observations. (Note that though the agent is building up a description of the effects it has on the world, it does not need to have any explicit knowledge of itself or of the principle of cause and effect.) From this the agent develops expectations; i.e., it can make simple predictions of what will happen when it chooses its actions, such as the effects on the retina of moving the finger forward and of rolling forward and backward and rotating in place. These predictions are well defined and straightforward to learn; they are therefore assumed for our purposes to be built in or already learned, and can be noticed or filtered out as necessary. There were no problems like this in diss.

Among the agent's first concepts are extended actions, implemented as options [28]. Table 3 defines three options in terms of initiation condition ($I$), policy ($\pi$), and termination condition ($\beta$). Each column describes a value or relationship previously described. In these examples, both $I$ and $\beta$ are binary, and therefore initiation and termination occur conclusively. These options—*press*, *press hard*, and *push*—are applicable whenever the agent's TOUCH sensor is activated, i.e., generally its finger is in contact with something. Each of these options simply causes the agent to move the finger forward until the force on the arm reaches a certain value, or, in the case of *push*, once motion other than the agent's own finger is detected visually (i.e., SEE CHANGE) or the touch sensor is no longer activated.

Notice that there are mathematical operations in both the $I$ and the $\beta$ column. These should be read as & ("and") and | ("or"). These operations are shown here for clarity, but would be implemented in the MLP as hidden units. In some cases in this article it makes sense to describe such relationships in the table where they are needed, as here; in other cases, it is more convenient to describe them with KU table entries.

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *press* | | TOUCH | *ff* | FORCE $> \Theta_1$ |
| *press hard* | | TOUCH | *ff* | FORCE $> \Theta_2$ |
| *push* | | TOUCH | *press* | **SC** \| **not** TOUCH |

Table 3: Early options for pressing and pushing. The threshold values are chosen such that $\Theta_4 < \Theta_3 < \Theta_1 < \Theta_2$, so that the agent can push something once there is sufficient force perceived on the finger, and the option will terminate once the force is too high ($> \Theta_1$), too low ($< \Theta_3$), or if the agent detects change visually.

**Real world, Layer 2.** In the real world, we develop expectations early on about what will happen if we take certain simple actions, and we build up extended maps of our immediate environment in terms of these expectations. Closing my eyes and slightly lifting my index finger from my computer's keyboard, I have a very good idea of what I will feel if my finger moves back again to where it was. But if instead of returning it, I move it a bit to the side, I still have an expectation of what will happen if I let it back down again. Thus I have something of a mental map of the keyboard beneath my fingers that is accessed simply by moving my finger around above the keys and imagining what would happen should I let them fall. That map is most obvious when I close my eyes, but it is more informative when my eyes are open. With closed eyes, we quickly lose track of geometrical relationships; with open eyes, the map is updated much more accurately. But the map is the same in both cases, and in both cases it is the map that informs us as to what our fingers will touch when they move. Our intuitive notion—that we simply *see* where our fingers are—is highly naive. Our visual observation is simply an image on the retina; what we interpret from that image are the spatial relationships encoded by the map.

**The Microworld, Layer 2.** The robot can reach out its finger to touch what might be in front of it. If it retracts its finger, it can predict what would happen if it were to reach out its finger again (though it may need to do this several times to establish the reliability of its signal; see Appendix A.1). This first piece of predictive knowledge is captured in Table 4 by the KU called TOUCHABLE (abbreviated **T**), which is a forecast about the agent's touch sensor if its finger moves forward. Again, TOUCHABLE is the nickname we have given the forecast as a mnemonic. Its true meaning is given by the table, not by the nickname.

Table 4 introduces two new columns, $c$ and $z$, which define the target for the forecast function (see Equation 1). In this table, all $c$ values are zero, and so each KU describes a prediction about the entry in the $z$ column at the time that the corresponding option terminates. This forecast can then immediately be used in further forecasts. Thus, the output column shows the unit's output, which in most cases is the forecasted quantity itself; i.e., the estimated value described by the *option*, $c$, and $z$ columns, according to Equation 1.

The map allows the robot's prediction for TOUCHABLE to be the same if it rotates slightly left and then back again to the right. This is captured by first predicting what TOUCHABLE should be after rotating left (TOUCHABLE LEFT) or right (TOUCHABLE RIGHT). The KU TOUCHABLE LEFT predicts what the value of TOUCHABLE will be after the robot rotates slightly to the left. Similarly, **TR** (the abbreviation for TOUCHABLE RIGHT) is the prediction of what the value of **T** will be after a 1° turn to the right. The agent can make a prediction about what will happen after turning slightly to the left and then back again to the right by making a prediction about the value of **TR** after a turn to the left (not shown).

The process can be generalized to cover the entire set of rotations available to the robot in either direction, one degree at a time. This *touch map* representation contains 360 forecasts, each compositionally related to the others. The KU TOUCH MAP($i$) serves as a prediction about whether the robot will touch something if it moves its finger forward after rotating $i°$ (where $-180 < i < 180°$). But more immediately **TM**($i$) is a prediction about **TM**($i \pm 1$) upon rotating 1° left or right. The KU **TM**(0) is a synonym for **T**; **TM**(1) a synonym for **TR**; **TM**(-1) a synonym for **TL**. In general **TM**($i$) is a prediction of **TM**($i-1$) for right rotations and **TM**($i+1$) for left rotations. (Note that the index $i$ is simply a notational convenience and not part of the machinery of the agent.)

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| TOUCHABLE | **T** | $\hat{f}$ | *ff* | 0 | TOUCH |
| TOUCHABLE LEFT | **TL** | $\hat{f}$ | $\curvearrowleft$ | 0 | **T** |
| TOUCHABLE RIGHT | **TR** | $\hat{f}$ | $\curvearrowright$ | 0 | **T** |
| TOUCH MAP $(0)$ | **TM**$(0)$ | **T** | | | |
| TOUCH MAP $(i), 0 < i <= 180$ | **TM**$(i)$ | $\hat{f}$ | $\curvearrowright$ | 0 | **TM**$(i-1)$ |
| TOUCH MAP $(i), 0 > i > -180$ | **TM**$(i)$ | $\hat{f}$ | $\curvearrowleft$ | 0 | **TM**$(i+1)$ |
| TOUCHABLE ADJACENT | **TA** | **TM**$(i) > \Theta$ for any $i$ | | | |

Table 4: Knowledge about the consequences of the agent's immediate actions, touch and rotate, organized into a map. The map is composed of 360 separate forecasts, 359 of which are forecasts of other map forecasts. The knowledge units **TM**$(0)$ and **TA** are not forecasts themselves. The first is simply a synonym for another forecast (**T**), and the second is a unit in the MLP that is a function of previously declared forecasts.

The result is a set of predictions describing what is immediately surrounding the robot. The map keeps track of the forecasts as the robot rotates and can therefore predict whether there is something that the robot can touch at every accessible point around its exterior. Building on the map, an important piece of knowledge can be condensed into one unit indicating whether there is something within reach anywhere surrounding the robot. This unit, **TA**, represents the forecast, "I can reach a situation where I will be able to get a high reading on my touch sensor by rotation only, or more succinctly, "I am adjacent to something touchable." It is a binary value: 1 if any of touch-map forecasts exceed a certain threshold, 0 otherwise.

**Real world, Layer 3.** We can use our internal maps to coordinate our actions. We can move a finger to the place where it will touch something or where it will press, say, the $T$ key. It is sometimes useful to consolidate these actions such that they are easily accessible. We can then think simply about pressing the $T$ key rather than planning a sequence of actions to achieve that goal. Furthermore, we can use our knowledge to set ourselves goals to learn about; for example, we can learn how to press the $T$ efficiently from whatever position our hands might start in.

**The Microworld, Layer 3.** In the microworld, options can be formed that refer specifically to knowledge units and make use of those predictive abstractions. A few examples are given in Table 5. The option *rotate left to touchable* can be used whenever the robot forecasts that it is adjacent to something touchable. It specifies that the agent should simply keep rotating left until its forecast of TOUCHABLE exceeds a threshold or the robot no longer believes it can achieve TOUCHABLE by rotation.

A different kind of option is defined by *rotate to touchable*, for which a policy must be learned that maximizes the target as defined by $c$ and $z$. That learned policy will consist only of rotations and will minimize the number of turns required to face something touchable. Since the robot has a complete map of what surrounding it is touchable, this policy is something that can in fact be learned.

| Option | abbrev | I | $\pi$ | $\beta$ | $c$ | $z$ |
|---|---|---|---|---|---|---|
| *rotate left to touchable* | *rlt* | **TA** | $\curvearrowleft$ | (**T** $> \Theta$) \| (1 - **TA**) | | |
| *rotate right to touchable* | *rrt* | **TA** | $\curvearrowright$ | (**T** $> \Theta$) \| (1 - **TA**) | | |
| *rotate to touchable* | *rt* | **TA** | maximize | **T** | $-\infty$ for $\uparrow$ and $\downarrow$ , -1 for turns | 0 |

Table 5: These options are based on the touch map and describe rotations until the agent's TOUCHABLE forecast is sufficiently high. In any practical system, the $\infty$ in the $c$column can be implemented by any sufficiently large value in comparison to $-1$, or alternatively the $-1$ can be replaced with a very small value.

**Real world, Layer 4.** In the real world, there are things that move and things that do not. Perhaps one of the first things that a baby learns about its environment is that some things are *moveable*, while others stay where they are and *push back*. As babies, we quickly learned to distinguish the former from the latter. So what exactly did we learn? If I push somewhere and I then see a change or I feel a reduction in the force against my hand, then I have perceived something special: I can now make an important distinction about the things in my world by predicting whether or not I will perceive this change when I touch something and push on it.

**The Microworld, Layer 4.** Among the agent's earliest knowledge units are those encoding movability and permanence. Table 6 shows three. The forecast MOVABILITY is defined in terms of the option *push*. Its target is defined by $z$, which is 1 when change is seen or when the robot's TOUCH sensor is no longer active (which occur when the robot pushes something that then moves). Thus, the KU measures the probability that, when the robot pushes with its finger, it will detect movement either visually or through disappearance of its TOUCH signal. This forecast should be high when the robot's finger is in contact with something moveable in the microworld.

PERMANENCE IN FRONT OF ME forecasts that the *push* option will end because the force against the finger has exceeded a certain level (which may occur when the robot is pushing against something that it cannot move).

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| MOVABILITY | | $\hat{f}$ | *push* | 0 | **SC** \| **not** TOUCH |
| PERMANENCE IN FRONT OF ME | **PIF** | $\hat{f}$ | *push* | 0 | FORCE $> \Theta_1$ |

Table 6: Knowledge units describing the immediate perception of things that will move and things that will not move when pushed.

The agent can now use these forecasts to keep track of what in its immediate vicinity will move if pushed and what will not. Just as with the touch map, the agent can also easily keep track of whether or not it is adjacent to PERMANENCE anywhere surrounding it. We can therefore assume at this point that every property that can be identified and verified through some test immediately in front of the agent can be generalized in the form of a map for all positions that the agent can rotate to. A map for PERMANENCE is shown in table 7, which also includes a KU for recognizing the situation in which PERMANENCE is not currently in front of the robot but can be achieved by rotation.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| PERMANENCE MAP $(0)$ | **PM**$(0)$ | **PIF** | | | |
| PERMANENCE MAP $(i), 0 < i <= 180$ | **PM**$(i)$ | $\hat{f}$ | $\curvearrowright$ | 0 | **PM**$(i-1)$ |
| PERMANENCE MAP $(i), 0 > i > -180$ | **PM**$(i)$ | $\hat{f}$ | $\curvearrowleft$ | 0 | **PM**$(i+1)$ |
| PERMANENCE ADJACENT | **PA** | **PM**$(i) > \Theta_{pa}$ for any $i$ | | | |
| CAN FIND PERMANENCE | **CFP** | $($**PIF** $< \Theta_{cfp})$ \| **PA** | | | |

Table 7: A map describing where in the robot's immediate vicinity it can expect to find PERMANENCE. A knowledge unit indicating whether PERMANENCE is adjacent to the robot. And a prediction that the agent can rotate to a position where it can find PERMANENCE in front of it.

And just as with the forecast TOUCHABLE, options can be formed for moving the agent with respect to the permanence map such that it faces the next closest permanence on its left or right, as shown in Table 8. These options can be initiated whenever PERMANENCE is not currently detectable in front of the robot but appears to be reachable by rotation; the options terminate when the robot no longer believes this condition to be true.

| Option | abbrev | I | $\pi$ | $\beta$ | $c$ | $z$ |
|---|---|---|---|---|---|---|
| *rotate left to permanence* | *rlp* | **CFP** | $\curvearrowleft$ | $(1-$ **CFP**$)$ | | |
| *rotate right to permanence* | *rrp* | **CFP** | $\curvearrowright$ | $(1-$ **CFP**$)$ | | |
| *rotate to permanence* | *rp* | **CFP** | maximize | $(1-$ **CFP**$)$ | $-\infty$ for $\uparrow$ and $\downarrow$ , -1 for turns | 0 |

Table 8: Options that rotate the agent to places where PERMANENCE is straight ahead.

These new options allow an alternative definition of **PA**, which is described in Table 9. This definition simply predicts whether the robot will ever find **PIF** by rotation alone. The ability to define the same piece of knowledge in multiple ways is a nice property allowed by GPK. In general, there is no reason to discourage the system from having multiple forecasts to obtain the same information. Each method is different. A different forecast function provides a different way for the forecast to be verified. Each method of verification may be best in different circumstances.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| PERMANENCE ADJACENT, METHOD 2 | **PA** | $\hat{f}$ | *rp* | 0 | **PIF** |

Table 9: Alternative definitions of knowledge.

### 3.3 First time, then Space

To encode individual objects and their spatial relationships, the agent must develop a spatial representation of its world. Just as you and I know where things are, the agent knows where things are in the sense that it knows how to get to states where it can interact with them. Thus, the agent represents the spatial relationships between things in its environment by knowing how to get from one to the other.

**Real world, Layer 5.** The spatial relationship between myself and say, the wall to my left, consists of both distance and direction. To get to the wall I must first orient myself toward it and then move forward over a certain distance. Distance in turn can be described predictively by the amount of time it takes to move from one place to another while following a certain policy.[16] Spatial relationships between arbitrary parts of the world can be described in the same way: if I first go to place A, how do I get to place B? It might seem simpler to measure the distance with a ruler, or, framed predictively, estimate what I would read if I were to stretch a ruler from one place to the other. But this conception of distance as a measurable phenomenon already relies very heavily on our intuitive understanding of space and spatial relationships, which in turn depend on our ability to interact with things in the world, such as by stretching a ruler from one place to another. The simplest, most intuitive notions of distance may be based on the predicted movements of ourselves or body parts from one place to another: if I touch the book with the tip of my finger and then move my finger along the most direct path toward my computer, how much time will it take?

**The Microworld, Layer 5.** The predictive agent can describe time predictively in two possible ways, both relying on the number of elapsed interactions with the environment. Say that the agent wishes to estimate the time elapsed while following an option from the current state $s_0$ to a particular termination state $s_d$. It could set $z$ to 1 when $s_d$ is reached, thus producing an event prediction as discussed in Section 2.2: the more interactions before $s_d$ is reached, the higher the chance of an alternative termination before then, and the smaller the forecast value $\hat{f}(s_0)$. Note that changes in the temporal resolution can be counterbalanced with corresponding changes to $\beta$: if $\beta = 0.01$ for $s \neq s_d$ when the sample rate is 10 Hz, then if the sample rate is increased to 100 Hz, $\beta$ can be set to (approximately) 0.001. The probability of termination *per second* would remain the same.

[16] Note that such spatial representations hold even for things that we cannot actually get to, such as the sun or the oval office, because we really only need to know how to get there in some way, even if that way is not possible. Thus, we tend to represent such things in terms that we understand, such as: if I could drive from here to the sun at 100 km/hr, it would take me 173 years to get there.

A different way to measure time is by directly counting the number of steps that elapse before reaching $s_d$. This can be done, for example, with a forecast where $c = 1$ (or any constant value) and $z = 0$ for $s \neq s_d$.[17] Just as in the previous case, all $c$ values can be adjusted to match the temporal resolution of the robot. Though either method can be used, the second method is in some ways more straightforward, especially for purposes of exposition, and so it is used heavily here.

The option in Table 10 describes the agent's movement from its current position to a place where it can detect permanence in front of it.

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *go forward to permanence* | *gfp* | $\mathcal{S}$ | $\uparrow$ | **PA** |

Table 10: This option takes the agent forward until it reaches a place where it can rotate to PERMANENCE.

Using this, the agent can build up estimates of the distance to the PERMANENCE immediately in front of it, measured as the number of time steps the robot requires to reach a place where it forecasts PERMANENCE IN FRONT OF ME. With that piece of knowledge, it is easy and convenient to build up a map of distances in all directions surrounding the robot, meaning that the robot can predict something about what it will see if it rotates in any amount and what it will predict from that visual information about the distance of PERMANENCE in its vicinity.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| DISTANCE TO PERMANENCE IN FRONT | **DPF** | | *gfp* | 1 | 0 |
| DISTANCE TO PERMANENCE MAP $(0)$ | **DPM**$(0)$ | **DPF** | | | |
| DISTANCE TO PERMANENCE MAP $(i), 0 < i \leq 180$ | **DPM**$(i)$ | $\hat{f}$ | $\curvearrowright$ | 0 | **DPM**$(i-1)$ |
| DISTANCE TO PERMANENCE MAP $(i), 0 > i > -180$ | **DPM**$(i)$ | $\hat{f}$ | $\curvearrowleft$ | 0 | **DPM**$(i+1)$ |

Table 11: Distances are described in terms of the number of forward steps the agent must take to reach a destination. In this case, the agent has reached its destination when the *go forward to permanence* option terminates.

With this map, the robot has some fairly sophisticated knowledge. The map contains 360 values that describe the robot's expectations about its forward motion. As it rotates in place, it updates the map based on its visual information. The map interprets that visual information and informs the robot how many time steps it would probably take for the robot to roll forward before reaching PERMANENCE (i.e., a state whereby the robot can sense something with its finger, but upon pushing with its finger it will observe that its FORCE sensor increases to a threshold but does not then decrease, nor is visual motion likely to be detected). And this information is then available in a complete circle surrounding the robot.

## 3.4 Objects and Things

What is the class of things we call "objects"? Most of the work in AI that deals with objects already assumes their existence and their epistemic primacy. Yet when building from the bottom up, this assumption is hard to justify. Babies are known to live quite happily—interacting with the world and learning from it—long before developing any strong notions of object. Our adult conceptualizations can easily mislead us when we try to reconstruct the world in terms of perception and action.

**Real world, Layer 6.** When we see a wall, a doorway or a room, we know what these things are just by looking at them. Long before we learned to recognize them at a glance, we first had to learn what they

[17] The value 1 is arbitrary, and can in principle be any value. A very low value might be chosen, for example, if the temporal resolution is high in order to keep the predictions roughly in the 0 to 1 range. Otherwise, the MLP would need to allow values in the range of the maximum cumulative value for any forecast in the system.

were (i.e., how we could interact with them). For example, walls primarily hinder our motion, not in just one place but in many adjacent places. Doorways have a certain shape and size, are generally connected to walls and (usually) allow us to get from or to a room. Rooms tend to be spaces enclosed by walls and a ceiling. All these things can be described in terms of how we can expect to interact with them. After years of such interaction, we become expert at recognizing them, in the sense that we become expert at making and updating our predictions about how we can interact with them based on what we see.

**The Microworld, Layer 6.** The robot sometimes has experience of PERMANENCE over a significant distance. It learns to predict that it can move forward with PERMANENCE beside it for a significant number of steps, which relies on the prediction of PERMANENCE beside it and on the ability to keep the PERMANENCE aligned next to it as the robot travels. Table 12 defines a few KUs that refer to the prediction that PERMANENCE can be verified on the robot's left or right side.

| Knowledge Unit (KU) | abbrev | output |
|---|---|---|
| PERMANENCE DIRECTLY TO MY LEFT | **PL** | **PM**(90) |
| PERMANENCE DIRECTLY TO MY RIGHT | **PR** | **PM**(-90) |
| PERMANENCE DIRECTLY BEHIND ME | | **PM**(180) |

Table 12: These units supply convenient names for refering to specific knowledge units aleady defined.

Two new options are introduced in Table 13 to help orient the robot and keep PERMANENCE aligned at its side. These are learned options that are applicable whenever the robot believes it is adjacent to PERMANENCE. The options rotate the robot as quickly as possible and terminate once PERMANENCE is aligned on its left or right side.

| Option | abbrev | I | $\pi$ | $\beta$ | $c$ | $z$ |
|---|---|---|---|---|---|---|
| *align permanence on left* | *apl* | **PA** | maximize | **PL** | $-\infty$ for $\uparrow$ and $\downarrow$ , -1 for turns | **PL** |
| *align permanence on right* | *apr* | **PA** | maximize | **PR** | $-\infty$ for $\uparrow$ and $\downarrow$ , -1 for turns | **PR** |

Table 13: These are learned options that adjust the orientation of the robot such that the closest PERMANENCE is directly left (or right) of the robot. The robot has to find the minimum number of rotations that will result in the desired alignment.

Using these two options, the robot can make predictions about the number of turns needed to perform the alignment. They are used by the knowledge units defined in Table 14, one that predicts the number of turns needed to align PERMANENCE to the robot's left and one for the corresponding prediction on its right. Though these provide redundant information, already available from the permanence map, they encapsulate the knowledge in a compact and convenient form, and, of course, as described above, redundancy is a source of robustness.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| PERMANENCE NEAR LEFT | **PNL** | $\hat{f}$ | *apl* | 1 | 0 |
| PERMANENCE NEAR RIGHT | **PNR** | $\hat{f}$ | *apr* | 1 | 0 |

Table 14: The agent describes how close PERMANENCE is to the left (or right) side by the number of rotations required to align PERMANENCE on that side

With these new knowledge units, options can be constructed for following PERMANENCE along while keeping it aligned on the left or right side. Table 15 defines two options, *follow permanence left* (*fpl*) and *follow permanence right* (*fpr*), which describe the motion of the robot rolling forward when possible and otherwise attempting to realign PERMANENCE on its left or right with a small number of rotations.

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *follow permanence left* | *fpl* | **PNL**$< \Theta$ | $\uparrow$ if **PL**, else *apl* | **PNL** $> \Theta_2$ |
| *follow permanence right* | *fpr* | **PNR**$< \Theta$ | $\uparrow$ if **PR**, else *apr* | **PNR** $> \Theta_2$ |

Table 15: These options allow the robot to roll forward while keeping PERMANENCE aligned at its left (or right) side. The policy is simply a forward roll when PERMANENCE is aligned and a realignment when not. Termination occurs when the robot predicts that an alignment can no longer be done in a small number of steps. (In this and other tables, all threshold values marked $\Theta$ are independent of each other unless stated otherwise.)

At last, the robot is ready to describe something predictively that we might call an "object." It now has the knowledge necessary to distinguish something *we* might describe as a "wall." The agent does not know the word "wall" or know any of our expectations about the word "wall." Nevertheless, the words "wall" and "object," are used in this article, generally without quotes, to refer to what the agent knows in terms more familiar to us, though this is merely a convenience, and we must keep in mind that it is a rough, illustrative correspondence.

The robot describes a WALL as PERMANENCE that it can roll alongside (either on its left or right) for a sufficient number of steps.[18] In the microworld it is useful for the robot to know how much curvature the PERMANENCE has. A PERMANENCE with a high degree of curvature may indicate something more like an island than a wall: a big rock, for example, is something the robot could roll alongside (in circles) indefinitely, which would have different significance to the robot. {It would be easy to add some KUs and options for interacting with rocks and islands, if you think that would be interesting.} The robot can distinguish these two types of interaction using an estimate of a counting relationship between forward steps and turns. Table 16 introduces two knowledge units, **WL** and **WR**, that describe a WALL on the robot's left or right in exactly this way, using *fpl* and *fpr*. Forward steps generate a positive value for $c$, while turns generate a (possibly different) negative value. If the predicted value (i.e., the expected sum of future $c$) exceeds a threshold, $\Theta$, then the robot predicts it can follow the PERMANENCE along a sufficient distance without turning too much. The robot will discount predictions about distant parts of the WALL relative to nearby parts, because the underlying options have a non-zero chance of termination at any point in the future, and it takes longer to reach more distant parts. Thus the straightness of a WALL nearby outweighs its possible curvature farther away.

This deeply layered description of the agent's possible pattern of interaction is defined completely in terms of low-level experience. It is purely predictive, relative to the agent's perspective, and therefore completely verifiable. Note that these forecasts will take on high values when the robot is nearby any straight, extended barrier in the microworld tall enough to impair forward motion (which is probably fairly low for the robot here). But it should be clear that forecasts with more subtle predictions can be added, allowing the agent to interact with and distinguish many different kinds of PERMANENCE that it might find in its environment.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| WALL ON LEFT | **WL** | $\hat{f} > \Theta$ | *fpl* | $k_1$ for $\uparrow$ , $-k_2$ for turns | 0 |
| WALL ON RIGHT | **WR** | $\hat{f} > \Theta$ | *fpr* | $k_1$ for $\uparrow$ , $-k_2$ for turns | 0 |
| WALL ON LEFT (BEHIND) | **WLB** | $\hat{f} > \Theta$ | *fplb* | $k_1$ for $\downarrow$ , $-k_2$ for turns | 0 |
| WALL ON RIGHT (BEHIND) | **WRB** | $\hat{f} > \Theta$ | *fprb* | $k_1$ for $\downarrow$ , $-k_2$ for turns | 0 |

Table 16: A straight PERMANENCE (called a "wall" here) on the agent's left or right is distinguished by the degree to which the agent can move parallel to it without turning too much. The threshold $\Theta$ is chosen relative to the values of $k_1$ and $k_2$ to distinguish PERMANENCE with a large degree of curvature from PERMANENCE with relatively little.

[18] Stated more precisely, the KUs having to do with WALL are all predictions that the robot will be able to roll forward a sufficient number of steps while simultaneously predicting the ability to detect permanence on its left or right side.

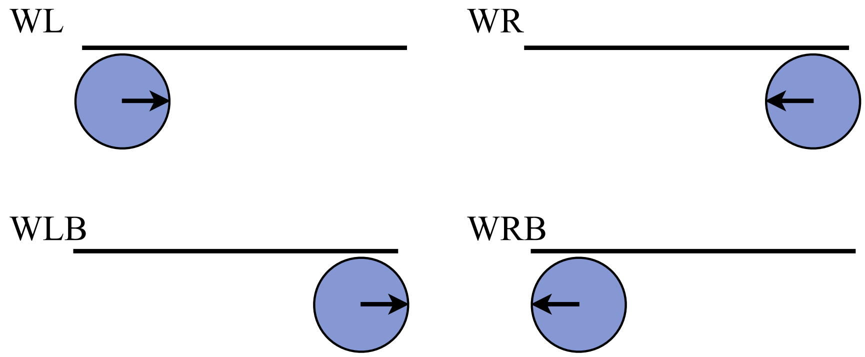


Figure 1: The robot is shown adjacent to a wall in 4 different canonical positions. In each position, a specific forecast can be made about what the agent could encounter (and verify) should it choose to do so. These forecasts (**wl**, **wr**, **wlb**, and **wrb**) are shown in each case and refer to their definitions in Table 16.

Units **wl** and **wr** only forecast a WALL that stretches out in front of the robot for some distance, but a WALL might also stretch out behind the robot instead, and knowledge units can be defined to capture this fact as well (see Figure 1); these are shown in Table 16 as **wlb** and **wrb**, which are much like **wl** and **wr**, but make predictions about backwards motion. They rely on two options, *fplb* and *fprb*, which are not shown, but which are the backwards-moving analogs of *fpl* and *fpr*. Equivalent, redundant, yet different definitions for *fplb* and *fprb* (also not shown) could work by instead predicting whether a WALL would be predicted should the robot first rotate 180°.

The four KUs of Table 16 describe forecasts that can be made from four canonical positions. The robot can now make forecasts about its ability to reach these canonical positions. In doing so, it forecasts how it can interact with the environment from nearby states. Table 17 is a map of the WALL relationships surrounding the robot. The map rests on the robot's ability to reach the canonical WALL states by rotation. The WALL map is a bit more complicated than the agent's other maps because the verification of WALL rests upon the verification of PERMANENCE, yet the WALL verification occurs while the robot is moving at a 90° angle from the PERMANENCE verification. Furthermore, the PERMANENCE verification can be made on both the left and right sides of the robot, and the WALL can extend in either or both directions away from the robot, as shown in Figure 1.

The WALL map is first defined in terms of WALL predictions on its left and right side **wm**(-90) and **wm**(90), and it is these that the rest of the map ultimately rests its predictions on, in a chain-wise fashion as for the previous two maps. Interestingly, it is only through the forecasts provided by the map that the robot can recognize WALL directly in front of it (**wif**), though it may do so visually (as described in previous layers). Thus, the visual image from the robot's retina must inform the forecasts of the map, which predict observations the robot could make were it to rotate 90° to the side, which in turn make some predictions (among others) about what the robot could sense with its finger were it to rotate 90° back again. And through this complex interaction, the robot recognizes the possibilities it has for interacting with the WALL in front of it.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| WALL MAP (-90) | **WM**(-90) | **WL** $\mid$ **WLB** | | | |
| WALL MAP (90) | **WM**(90) | **WR** $\mid$ **WRB** | | | |
| WALL MAP $(i), i \in \{-179, \ldots, -91, 0, \ldots, 89\}$ | **WM**$(i)$ | $\hat{f}$ | $\curvearrowleft$ | 0 | **WM**$(i+1)$ |
| WALL MAP $(i), i \in -\{-89, \ldots, -1, 91, \ldots, 180\}$ | **WM**$(i)$ | $\hat{f}$ | $\curvearrowright$ | 0 | **WM**$(i-1)$ |
| WALL IN FRONT OF ME | **WIF** | **WM**(0) | | | |
| WALL IN BACK OF ME | **WIB** | **WM**(180) | | | |
| WALL ADJACENT | **WA** | **WM**$(i) > \Theta$ for any $i$ | | | |

Table 17: The WALL map is more complex than previous maps. It is defined by verifications of WALL on its left (or right) side while the agent travels alongside the WALL. Verification is done by touch, where the robot rotates 90° to the left or right—**WM**(-90) or **WM**(90).

The map of Table 17 also shows that the way something is verified (the specification of the forecast function) is not necessarily the way the information is best obtained. For example, although the predicted value for **WLB** is described in terms of the robot moving backwards along the WALL, the robot's camera, which is the device that supplies the most information about the robot's whereabouts, faces forward. But it may be the case that the forecast can best be estimated if the agent turns 180° degrees. So, though the robot may not explicitly forecast, "if I turn 180°, **WL** will be high," it may be that the best way to gather evidence for **WLB** is to turn around and take a look. Similarly, the best way to verify that the car key is in your pocket might be to take it out to the car and try it, but this generally isn't necessary, because you can obtain fairly high confidence in your prediction by gathering other evidence. Also note that **WIF** and **WIB** are defined as predictions based on an initial left rotation but could just as efficiently be based on a right rotation.

**Real world, Layer 7.** What is the difference between a man and a woman, a diamond and paste, or a wall and a window? They may seem similar in many ways, but in each case, there is some essential quality that makes them different. One quality that makes a window different from a wall is its transparency. But what does that mean in predictive terms?

**The Microworld, Layer 7.** A window is something that the robot can interact with like a WALL, but is different somehow. One clue might be that if the robot faces the window it can sometimes observe things moving on their own, and one way to tell if something is moving on its own is simply to stop moving and watch. Table 18 introduces the *look and see* option, in which the agent takes no outward action in the world but simply watches until termination, which has a small chance of occurring at every step.

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *look and see* | *las* | $\mathcal{S}$ | $\varnothing$ | 0.01 |

Table 18: This option takes the *null* action (the action that has the minimum impact on the robot's observations) for a variable amount of time. The probability of termination at every step is 1%. Thus, the agent can use this option to estimate the likelihood of events that may occur over a fairly short period while the agent stands motionless.

A good policy for gaining information is simply remaining still and observing. In the very simplest case, this policy can reveal whether activity is occurring external to the agent. In Table 19 the knowledge unit THINGS ARE HAPPENING HERE measures the probability that the agent will SEE CHANGE if it does nothing. Using this prediction, it can distinguish WINDOW IN FRONT OF ME from WALL IN FRONT ME by the probability it will observe changes in its visual field when it faces one (WINDOW IN FRONT OF ME) .

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| THINGS ARE HAPPENING HERE | **THH** | $\hat{f}$ | *las* | **SC** | 0 |
| WINDOW IN FRONT OF ME | | **WIF** & (**THH** $> \Theta$) | | | |

Table 19: WINDOW IN FRONT OF ME is the prediction of seeing change while facing a WALL and doing nothing.

**Real world, Layer 8.** We can tell where we are in relationship to the things around us largely because we keep track of where things are relative to where we are. This means keeping estimates of roughly how far things are from us. If you stand up, close your eyes, and rotate around slowly in a familiar place, you can predict not just what you are facing, but about how far away those things are. If you were to stand in the center of a large room (5 meters $\times$ 5 meters, say), close your eyes, turn around 180°, open your eyes and suddenly see a wall only a few centimeters in front of your nose, you would be very surprised. We know the distances to the things around us because we keep track of them; we have a model of our surroundings and we update that model continually.

**The Microworld, Layer 8.** The robot measures distance predictively in terms of the number of steps it takes to get from one place to another. The option in Table 20 takes the robot straight ahead until it forecasts WALL ADJACENT (a prediction that by rotation alone it can reach a state where it will forecast WALL IN FRONT OF ME).

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *go forward to wall* | *gfw* | $\mathcal{S}$ | $\uparrow$ | **WA** |

Table 20: This option moves the robot forward until it reaches a WALL. It may contact the WALL at any angle.

The knowledge units in Table 21 use the *go forward to wall* option to describe the distance to WALL in front of the robot and then to generalize this to an abstract map of distances to all states surrounding it (i.e., states it can reach with the *roll forward* action) and where it will forecast **WA**, just as was done with the permanence map of Table 11. The information in the WALL map and PERMANENCE map will frequently be the same.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| DISTANCE TO WALL IN FRONT | **DWF** | $\hat{f}$ | *gfw* | 1 | 0 |
| DISTANCE TO WALL MAP (0) | **DWM**(0) | **DWF** | | | |
| DISTANCE TO WALL MAP $(i), 0 < i \leq 180$ | **DWM**$(i)$ | $\hat{f}$ | $\curvearrowright$ | 0 | **DWM**$(i-1)$ |
| DISTANCE TO WALL MAP $(i), 0 > i > -180$ | **DWM**$(i)$ | $\hat{f}$ | $\curvearrowleft$ | 0 | **DWM**$(i+1)$ |

Table 21: Distances to nearby WALL forecasts are described exactly as for PERMANENCE (Table 11): in terms of the number of forward steps it will take the agent to reach one. Based on the forecasted distance in the forward direction, the agent maintains a map of forecasts for distances to **WA** in all directions surrounding the robot.

**Real world, Layer 9.** Because we can look around and keep track of what we see, we can form a picture of what is in our vicinity without needing to see everything at once. We know we are in a room even though we cannot see all four walls at the same time. We know whether we are in a large or a small room by knowing what the relative relationships are between the walls.

**The Microworld, Layer 9.** The robot now has enough knowledge to begin to understand the spaces surrounding it. It can create new forecasts to distinguish certain states (which we will call "rooms") where it forecasts that it can move directly to WALL on four of its sides. Figure 2a shows a situation in the microworld where this might occur.

The robot can characterize and distinguish different kind of ROOMs by using the distance relationships it has built up so far. For convenience, a few specific pieces of knowledge are given names in Table 22.

| Knowledge Unit (KU) | abbrev | output |
|---|---|---|
| DISTANCE TO WALL ON LEFT | **DWL** | **DWM**(-90) |
| DISTANCE TO WALL ON RIGHT | **DWR** | **DWM**(90) |
| DISTANCE TO WALL BEHIND | **DWB** | **DWM**(180) |
| LATERAL LEEWAY | **LL** | $\mathbf{DWL} + \mathbf{DWR}$ |
| ANTEROPOSTERIOR LEEWAY | **AL** | $\mathbf{DWF} + \mathbf{DWB}$ |
| LATERALLY CENTERED | **LC** | $(\mathbf{DWL} - \mathbf{DWR} < \Theta_{\mathbf{LC}})$ & $(\mathbf{DWR} - \mathbf{DWL} < \Theta_{\mathbf{LC}})$ |
| ANTEROPOSTERIORLY CENTERED | **AC** | $(\mathbf{DWF} - \mathbf{DWB} < \Theta_{\mathbf{LC}})$ & $(\mathbf{DWB} - \mathbf{DWF} < \Theta_{\mathbf{LC}})$ |

Table 22: A few convenient names to simplify later definitions.

Table 23 describes some fairly high-level knowledge. Several of the KUs in this table assume that the forecasts are estimated from a canonical state, illustrated in Figure 2b, where the agent predicts that: (1) it can move directly to WALL on all four sides; (2) a slight rotation to either side will increase its forecasted distance to a WALL in front; (3) the amount it can move left or right before reaching WALL is not greater than the amount it can move forward or backward. (How the agent can use these KUs to get to the canonical position is described in Layer 11.) From these canonical states the robot can make reliable predictions about reaching WALL forecasts nearby; it can thereby estimate distances in a consistent way. The KU ROOM AREA is the robot's total estimated distance to the WALL in front and in back of it times the total distance to the WALL on both sides. It has a high value when the robot forecasts large distances to WALL surrounding it, and it has a low value when the robot forecasts that it can reach a WALL in every direction relatively quickly. It is useful for characterizing many different states of the microworld.

The HALLWAY KU has a high value when the robot forecasts that it can get to WALL on its sides rather easily but will take a while to get to WALL in front and back of it (e.g., when the robot is in a long, narrow space). The MIDDLE OF A HALLWAY KU predicts that the robot's distance to WALL on its left is about the same as its distance to WALL on its right, while it will take considerably longer to reach the WALL in front or back of it (e.g., Figure 2c). The CENTER OF ROOM KU indicates that the robot's estimated distance to WALL in front and back of it are about the same as well. The SMALL ROOM KU has a high value when ROOM AREA is below a certain threshold. The LARGE ROOM KU is similar to the SMALL ROOM KU, but the threshold is higher and ROOM AREA must be greater.

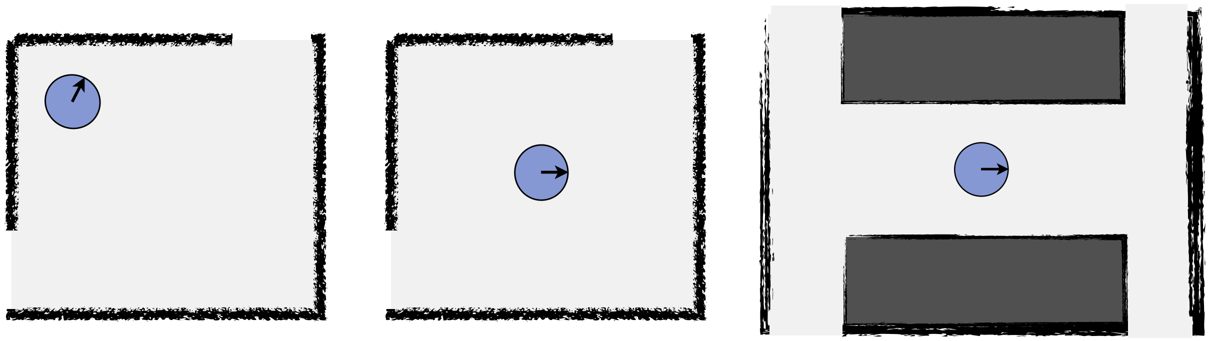

Figure 2: (a) The robot is shown in a microworld room, where it might forecast that it can find a wall on four different sides (using knowledge unit $\textsc{r}$ from Table 23). The robot is shown in a canonical position in (b) a room and (c) a hallway. In the canonical position, the robot may forecast its detection of walls directly to its left and right, as well as directly ahead and behind it.

| Knowledge Unit (KU) | abbrev | output |
|---|---|---|
| ROOM | **R** | $(\textbf{DWF} < \Theta_{\textbf{R}})$ & $(\textbf{DWL} < \Theta_{\textbf{R}})$ & $(\textbf{DWR} < \Theta_{\textbf{R}})$ & $(\textbf{DWB} < \Theta_{\textbf{R}})$ |
| ROOM (CANONICAL POSITION) | **RCP** | $\textbf{R}$ & $[\textbf{DWM}(-1) > \textbf{DWM}(0) < \textbf{DWM}(1)]$ & $(\textbf{LL} \leq \textbf{AL})$ |
| HALLWAY | **HW** | $\textbf{RCP}$ & $(\textbf{LL} < \Theta_{\textbf{HW}_1})$ & $(\textbf{AL} > \Theta_{\textbf{HW}_2})$ |
| MIDDLE OF HALLWAY | **MHW** | $\textbf{HW}$ & $\textbf{LC}$ |
| CENTER OF ROOM | **CR** | $\textbf{LC}$ & $\textbf{AC}$ |
| ROOM AREA | **RA** | $\textbf{LL} \times \textbf{AL}$ |
| SMALL ROOM | **SR** | $\textbf{RA} < \Theta_{\textbf{SR}}$ |
| LARGE ROOM | **LR** | $\Theta_{\textbf{SR}} < \textbf{RA} < \Theta_{\textbf{LR}}$ |
| ROOM AREA, VARIANT 2 | **RA**2 | $\sum_{i=0}^{360} \textbf{WM}(i)$ |

Table 23: These new knowledge units, describing predictions the robot can make about its WALL forecasts, might be useful in various areas of the microworld. Several of these rely on the robot finding its way to a canonical state (**RCP**), illustrated in Figure 2.

Of course, few rooms are perfect rectangles, and it is not the intention here to imply that they should be, or that the robot can only make predictions about rooms that are rectangular. The knowledge unit **RA**2 captures a completely different way of estimating the amount of space surrounding the agent, taking into account all the forecasts in the agent's WALL map. Similarly, distance can be measured in different ways (e.g., taking out a ruler and measuring), and the method used here is only the most fundamental and rudimentary, but the same techniques demonstrated above can be used to build other forecasts that distinguish other shapes of rooms and other ways of measuring, and though they probably also have very interesting predictive descriptions, the length of this article is also an important measurement.

The KUs from these last paragraphs describe fairly generic knowledge in terms that are intuitive for us and seem nearly objective—distinguishing things that resemble some of our own concepts and categories—yet it is important to remember that *all* of these definitions are constructed purely out of predictions of future experience, based entirely on interaction with the environment. They have been designed for this paper to convey a feeling for the power of the representation. For a continual-learning agent, the path through discovery of complex knowledge would likely lead to the construction of different kinds of forecasts and different ways of interacting with the regularities in the environment.

**Real world, Layer 10.** Once we have a clear understanding of an object, such as a key or a room, we can describe our relationship to it. We can, for example, consider whether we could find a key in our pocket, or to what extent we are in the middle of a room or a hallway. And though the canonical test for a key may be to put it into the ignition and turn it, it is nevertheless useful to make predictions regarding the likelihood that the test would succeed, even if the car is not within walking distance.

**The Microworld, Layer 10.** Once the agent has KUs that identify certain canonical situations, that knowledge can be referenced in subsequent forecasts. For example, the robot can predict the number of steps it will take and the likelihood of arriving in the canonical position from any other states.

Table 24 shows a set of options that take the robot to a canonical state, all learned through maximizing $c$ and $z$. In the first case, the robot, already in a ROOM, rotates until it reaches an orientation meeting the criteria defined by $\beta$: it is facing a WALL directly, (meaning that its estimated distance to the WALL will increase if it rotates a bit to the left or the right), and the distance between the WALL forecasts at its sides is not more than the distance between WALL in front and back, thus potentially leading to the orientations shown in Figures 2b and 2c. The next two options take the agent to a state where its KUs MIDDLE OF A HALLWAY and CENTER OF A ROOM both have values of 1.

| Option | abbrev | I | $\pi$ | $\beta$ | $c$ | $z$ |
|---|---|---|---|---|---|---|
| *rotate to room (canonical position)* | *rrcp* | **R** | maximize | **RCP** | 0 for turns, else $-\infty$ | 1 |
| *go to middle of hallway* | | **HW** | maximize | **MHW** | 0 | **MHW** |
| *go to center of room* | | **R** | maximize | **CR** | 0 | **CR** |

Table 24: These options are based on the generic ROOM descriptions given in Table 23. The first allows meaningful use of those prediction by establishing a canonical orientation with respect to a ROOM. The others take the agent into states that have high values for the MIDDLE OF A HALLWAY and CENTER OF A ROOM KUs.

The agent can use these options for describing its predictions with respect to the DOOR and HALLWAY KUs. Table 25 shows new KUs that help the robot forecast what it will perceive upon rotating to a canonical room position (**RCP**), in particular, whether it will then forecast WALLs surrounding it and what relationships those WALL forecasts might have.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| I'M IN A ROOM | **IR** | **R** | | | |
| I'M IN A SMALL ROOM | **ISR** | $\hat{f}$ | *rrcp* | 0 | **SR** |
| I'M IN A LARGE ROOM | **ILR** | $\hat{f}$ | *rrcp* | 0 | **LR** |
| I'M IN A HALLWAY | **IHW** | $\hat{f}$ | *rrcp* | 0 | **HW** |

Table 25: These knowledge units establish relationships between the agent and the ROOM knowledge units, which make predictions about WALL forecasts.

With the new KUs from Table 25, the robot can build options that will terminate upon reaching a situation where it believes it can reach one of the canonical positions (**RCP** or **HW**) by rotation alone.

| Option | abbrev | I | $\pi$ | $\beta$ |
|---|---|---|---|---|
| *go forward into small room* | *gfr* | $\mathcal{S}$ | $\uparrow$ | **ISR** |
| *go forward into hallway* | *gfh* | $\mathcal{S}$ | $\uparrow$ | **IHW** |

Table 26: These options take the agent to states where it forecasts that it can reach high values for two of the KUs in Table 25

With the new options from Table 26, the robot can predict distances to any of the states currently reachable by these options. Then, using the same techniques for the WALL and PERMANENCE maps shown


Figure 3: A doorway's appearance leads to certain expectations as to its functionality.

above, the robot can thus create a 360° distance map for all ROOM and HALLWAY states that it can reach through forward motion.

| Knowledge Unit (KU) | abbrev | output | option | $c$ | $z$ |
|---|---|---|---|---|---|
| ROOM IN FRONT | | $\hat{f}$ | *gfr* | 0 | 1 |
| HALLWAY IN FRONT | | $\hat{f}$ | *gfh* | 0 | 1 |
| DISTANCE TO ROOM IN FRONT | **DRF** | $\hat{f}$ | *gfr* | 1 | 0 |
| DISTANCE TO HALLWAY IN FRONT | **DHF** | $\hat{f}$ | *gfh* | 1 | 0 |
| DISTANCE TO ROOM MAP | **DRM** | *as with previous maps* | | | |
| DISTANCE TO HALLWAY MAP | **DHM** | *in tables above* | | | |

Table 27: These forecasts predict the agent's ability to roll forward to a state where the ROOM and HALLWAY KUs have high values, and the number of steps required to do so. Distance maps for I'M IN A ROOM and I'M IN A HALLWAY can be defined exactly as was done for other distance maps above.

**Real world, Layer 11.** Sometimes we recognize things by their appearances, sometimes by their functionality. Doorways have both a functionality and an appearance, which can lead to some confusion, as shown in Figure 3. In general, we expect doorways to provide passage into, out of, or between rooms.

**The Microworld, Layer 11.** The agent now has a large repertoire of sophisticated forecasts that it can use to make sense of more subtle aspects of the things in its surroundings and their relationships. Table 28 shows a few convenient definitions that make it easy to build forecasts for interacting with a DOORWAY. The KU describes a condition that is only met when the robot finds itself in a certain canonical situation in which it can perceive PERMANENCE nearby on each side and a ROOM immediately in front of it. In the microworld the agent will generally only make this forecast when it stands in or passes through a narrow passageway between rooms.

| Knowledge Unit (KU) | abbrev | output |
|---|---|---|
| DISTANCE TO PERMANENCE LEFT | **DPL** | **DPM**(90) |
| DISTANCE TO PERMANENCE RIGHT | **DPR** | **DPM**(-90) |
| DOORWAY (CANONICAL POSITION) | **DW** | (**DPL** + **DPR** $< \Theta_1$) and (**DRF** $< \Theta_2$) |

Table 28: A DOORWAY is a canonical situation in which the robot can perceive PERMANENCE on each side and a ROOM in front of it. The agent can refer to this canonical situation in future options and knowledge units.

### 3.5 Building knowledge at a larger scale

Besides its immediate vicinity, it is also important for an agent to represent knowledge about things that are farther away, both in space and time.

**Real world, Layer 12.** If you have ever been to the Eiffel tower, you know roughly its shape and size and more or less how it is laid out. You no longer need to be nearby in order to talk about its structure. Nevertheless, if you go back there again, you will be able to verify (or not) what you expect it to look like. In more mundane situations, we can distinguish the kitchen from the living room by how things are laid out in each. Even without the furnishings, the locations of the walls and doorways are important clues to us.

**The Microworld, Layer 12.** The agent can also represent knowledge about states that might be currently hard to reach. It can represent relationships about places that are far away, and it can then use this knowledge when it returns. Places in its environment have certain relationships; forecasts for WALL, DOORWAY, and HALLWAY can be made from certain states with specific relationships to each other. The agent might characterize these relationships with new KUs: a LIVING ROOM, for example, might be a set of states in which the LARGE ROOM forecast can be made and where, if it stands in one DOORWAY, it will forecast another DOORWAY at about 40° in its DOORWAY map, reachable in about 50 time steps. If the robot rolls through that second DOORWAY, its I'M IN THE KITCHEN forecast may make similar predictions, forecasting how the robot can move between different states or canonical positions that it can now easily reach (e.g., of WALLS and DOORWAYS, and states where the DOORWAYS lead). The robot can describe actions now in very high-level, geographic terms: it can build options that take it to states that are characterized in this way (e.g., KITCHEN or LIVING ROOM). This general process can continue until it can make predictions about how it can get to, and what it will find in, every part of its house. And all these predictions are described in terms that can be verified in the sensorimotor stream.

Just as the robot can make sophisticated, high-level, long-term predictions about its own house, it can also make predictions about different houses and the physical relationships between them. Yet "house" here is really only a metaphor. It represents in the abstract: (a) anything that can be distinguished by its unique pattern of interaction in the sensorimotor stream, and (b) the relationships that can be discovered between those things, which in turn become a pattern of interaction at a higher level. This is continual learning, the constant process of constructivism: the process of identifying patterns of interaction, encapsulating them into forecasts and options, and then using these as the starting point for ever more sophisticated knowledge units and options.

This section has told a specific and concrete story in which an agent's knowledge was built up rigorously in twelve layers, from senses and actions upwards. The next two sections discuss a variety of different kinds of issues relating to predictive representations. These sections rely on the understanding gained here to paint (in less detail) a broader picture.

## 4 Agent, State, and Predictive Knowledge

Our experience of the world is largely, though often invisibly, informed by our implicit, conditional predictions. As you read this, you are reading it somewhere, in some environment around you. Your eyes are on the page (or screen) but you are surprisingly well aware of your environment in many ways. You probably know what you would see if you were to turn around and look behind you. Perhaps there is a bookshelf there, and if so, you know that if you were to turn around, you would see books. If you were to see something completely different, say an African savana, you would be very much surprised. But this prediction is conditional: you know that you would see the bookshelf if you were to turn around, but you are under no obligation to turn around. Your conditional prediction is present without necessitating verification. Yet it is more than just a prediction that you *can* make: the prediction that you *could* turn around and *would* see a bookshelf and not a savanna is a subtle part of your current experience of reading this text, slightly coloring and influencing it. If something completely different were behind you, say a vertical drop of 1000 meters; the knowledge that you could lean over backwards and fall to your death would color your experience, even if what you actually observe and perceive while reading this text in that case would be identical to what you are observing and perceiving right now.

In the same way, you know what to expect from interactions with objects near you. You know what will happen if you twist your pen in your finger tips: it will gently turn, maintaining a certain kind of pressure and texture on your fingers and a certain kind of consistent image on your retina; it will not burst into feathers and fly away. If rolling the pen could cause it to blossom into a pidgeon, you would handle it differently, and that difference would be informed by and due to your implicit prediction of what might result otherwise.

An important hypothesis of this paper is that our understanding and knowledge of the world, including our place in it, is nothing more nor less than the full set of predictions we constantly make about what we would observe should we take certain actions.

## 4.1 Representing State

Because the term "state" is sometimes ambiguous, we use it in its purely Markov sense: the state encapsulates a system's behavior over all possible futures. In other words, the state encodes a probability distribution over the full tree that spans all possible futures the system might encounter. A representation of state may not be easily interpretable, and the probabilities might only be discoverable through testing and experimentation. A thermostat, for example, might only through interaction and experimentation reveal its state, including whether the thermostat currently has the heat turned on or off, how long the heat has been on or off, and what the thermostat's current set point is.

Each unique state describes a different probability distribution over possible futures but does not necessarily identify a unique past, because the same state can result from different histories. In fact, it is not a requisite of state to say anything at all about the past, only about the future.

Both the agent and environment are systems, and it makes sense to talk about the state of each. From the agent's perspective, the state of the environment must be estimated in order to make predictions about the future for purposes of planning and action selection. Therefore, the word "state" can potentially mean (at least) three different things: the state of the agent, the state of the environment, and the agent's estimate of the state of the environment. Unless stated otherwise, "state" here ($s \in \mathcal{S}$) refers to the first of these, the state of the environment; and the third of these, the agent's estimate of $s$ is always called the agent's "state vector," $\mathbf{s} \in \mathbb{R}^n$, where $n$ may change over time as forecasts are added or removed.

Because the state vector encodes everything the agent can predict about how it might affect the environment, the way in which the agent places a value on the future is (in principle) completely reducible to the way that it evaluates its state vector. The way that it values its state vector implicitly weighs every fork in the decision tree of its temporally unravelling future. Thus state, value, and prediction of the future, are all closely related and the role for the agent's state representation becomes clear.

The agent's state representation should help the agent predict the future.

## 4.2 Knowledge as State

It is our hypothesis that the agent's state is nothing more nor less than all of its knowledge, which is to say, all of its primitive observations together with all of its forecasts. Describing these as a vector:

$$\mathbf{s}_{t+1} \equiv o_{t+1} \circ \hat{F}(\mathbf{s}_t),$$

where $\hat{F}$ is the full set of individual forecasts $\hat{f}^i$, and the symbol $\circ$ concatenates two vectors. Together, this set of values encapsulates everything the agent knows about its world, describing exactly everything it can predict about how the world might respond to the agent's activity. This is one notable advantage of the predictive framework: its state representation clearly defines the full span of the agent's knowledge. However, what it can do with that knowledge is less explicit. For example, it can use that knowledge to imagine future scenarios and to plan, evaluate, and select actions.

## 4.3 Action Selection

A continual-learning agent gains knowledge by interacting with the world. As described so far, the predictive agent only learns to forecast what occurs in its environment as a response to its actions, but we have not yet discussed *how* it chooses one action over another. In the general reinforcement-learning scenario, agent's choose actions to maximize expected future discounted reward or utility. This expected future reward can be estimated either with a function approximator [25], by explicitly rolling out the tree of possible futures [8], or with a mixture of these methods [**?**]. In the first and third cases, a function approximator $\hat{V}$ learns to approximate the *value function*, $V : \mathcal{S} \rightarrow \mathbb{R}$, where the *value* of each state, $V(s)$, is the expected discounted future reward accrued while following the optimal policy $\pi^*$ starting from that state:

$$V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s = s_0, \pi^*\right],$$

where $\gamma \in [0, 1]$ is a discount factor. In each state, $s$, the optimal policy $\pi^*$ selects an action $a^*$ that results in the highest average discounted future reward:

$$a_t^* = \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}\left[r(s_t, a) + \gamma V(s_{t+1}) \mid s_t, a\right]. \tag{5}$$

Our hope is that by encoding its state explicitly as predictive knowledge, the agent will be able to choose good actions in many novel situations; i.e., the state representation will generalize well (some evidence to this effect has been shown for PSRs [16]). However, this hope is mildly complicated by the fact that forecasts do not predict what *will* happen—only what *could* happen. It might seem that $\hat{V}(\mathbf{s}_t)$ should be low for states in which there are forecasts of impending harm (outcomes associated with low or negative reward), and high for states with forecasts of beneficial outcomes (those with high or positive reward). But just because some of its forecasts indicate that the agent could easily and dramatically harm itself (e.g., while driving down the freeway, cutting vegetables in the kitchen, or waiting for the subway), these forecasts do not imply that those states are bad or even something to avoid. And a high forecast for one beneficial outcome of a behavior does not imply that the same behavior might not simultaneously produce other, more harmful outcomes. Furthermore, there is always a chance of failure for any behavior and forecast, and there may be (and generally always will be) forecasts for many beneficial and many harmful outcomes at the same time. Clearly, the value of the state should reflect the agent's ability to obtain benefits while avoiding harm.

How then should $\hat{V}(\mathbf{s}_t)$ be calculated? Notice that Equation 5 can also be written as a forecast and that the policy that maximizes this forecast is the agent's estimate of $\pi^*$.[19] This (non-stationary) policy, $\Psi$, is the one we expect the continual-learning agent to follow at all times,[20] but because $\hat{V}(\mathbf{s})$ is an approximation (and $\mathbf{s}$ is only an approximation of $s$, with new forecasts being added to $\mathbf{s}$ at different times), it might be very inaccurate in many novel situations, which could be dangerous to the agent.

So how can the agent obtain better results while avoiding danger? One method would be to have a forecast that predicts the probability of a drastic reduction in $\hat{V}(\mathbf{s})$ when following $\Psi$. This forecast might be a good indicator of danger, but would not without inspection reveal the reason for the prediction. (Similarly, a different forecast might predict the probability of a sudden improvement in $\hat{V}(\mathbf{s})$.) When there are both good and bad eventualities that might conflict (e.g., picking up a hot tea cup, asking for a raise, etc.), $\Psi$ can be enhanced with planning and imagination to help choose actions that maximize the good and minimize the bad. Planning and imagination are ways to play out possible scenarios, learn from them, and update $\hat{V}(\mathbf{s}_t)$ with more carefully considered values of future possibilities.

## 4.4 Imagination, Planning, and Reasoning with Predictive Knowledge

An agent that can predict the likely consequences of its behavior can choose how to behave so as to bring about the results it prefers. Thus, it is desirable for the agent's knowledge, its representation of state, to

[19] {not sure how to say this, since actually the $\gamma$ makes this different from a forecast, and also, there is no termination, or could one say that termination can always occur but $I = \mathcal{S}$ so it immediately starts up again?}

[20] We call it $\Psi$, because it is the agent's Policy of Self Interest

enable the agent to imagine the future under different scenarios; i.e., to *plan*.

Many of our human faculties that allow us to deal with abstract knowledge rely on our imagination. I may have seen a lion at the zoo and may have picked up a cat in my living room, but to predict what it would be like to lift a lion, I must use my imagination, because I have never had direct experience with it. Imagination is an important feature of predictive knowledge, not just because it allows an agent to simulate things that it has never directly experienced, but also because it can be used for reasoning and planning even in familiar situations.

It is not within the scope of this paper to detail specific mechanisms for imagination in predictive systems, but there are a few principles that need to be outlined here. First, a rudimentary ability to imagine is already present in the agent. If the predictive agent can, for example, close its eyes and take actions in the environment, it will update its forecasts based on the limited information it has available, thus imagining the changes that it cannot directly perceive in the environment due to its actions. (We humans can do the same thing; see Section 3, Layer 2). A more sophisticated sense of imagination probably requires a dedicated mechanism. Second, besides making predictions just from the current state vector, $\mathbf{s}_t$, forecasts also allow prediction from other state vectors $\mathbf{s}'$ (and their corresponding states, $s' \in \mathcal{S}$). Let us call these "imagined" state vectors. Thus, the agent may choose to alter its current state vector in certain ways and then see what the resulting forecasts would be from the imagined state: $\hat{F}(\mathbf{s}')$. If the agent has a model, $m : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, that estimates the next state vector given the current state vector and chosen action, then the agent can choose imagined actions and estimate the resulting state vector: $\mathbf{s}'' = m(\mathbf{s}', a)$. This process can clearly continue indefinitely. Since all the forecasts are contained within the state vector, the model also estimates the resulting forecasts.

The combination of model and forecasts allows simple but powerful planning, which results in better policies and action selection. On skis I am constantly predicting the possibility of crashing. That prediction is also associated with large negative reward. If the estimated probability becomes too great, I can search quickly for an action that I believe will reduce that particular probability. In the same way, forecasts provide focused knowledge for predicting the future with respect to those things that are of most concern to the agent (i.e., that impact its long-term receipt of reward).

This general one-step planning method is possible with most traditional models. But forecasts allow a far more powerful type of planning because they estimate option-dependent targets, and these targets have specific semantics. In the case of an event forecast (Section 2.2), which estimates the expected value of a specific observation or forecast unit for a given option, the agent can imagine the consequences of taking an option by copying all of the forecasts based on that option into their corresponding places in the state vector, yielding a new, imagined state vector. For example, if a humanoid agent has an option that moves its hand forward slowly until it touches something, and if it has forecasts to predict its joint angles upon termination of that option, then those forecasts can be substituted into the imagined state vector in the place of the actual joint angles, and the imagined state vector can then be used to generate more forecasts. Or, if the agent has an option that takes it to the kitchen and it has forecasts for its local map at termination of the option, then it can substitute all those forecasts into the state vector to build a picture of its surroundings at termination of the option. Or, to decide whether to put on a jacket before going outside, the agent needs to know how hot it will feel outside with a jacket on. Let us say it has no such specific forecast to examine, but it does have an option for putting on its jacket and a forecast for how hot it will feel upon going outside. To choose whether to put on the jacket now, it can imagine putting on the jacket (first option plus all forecasts related to the option) and then checking the second forecast to see how hot it will feel. Thus, planning can be used to predict the consequences of options and can occur at the same level of abstraction as the agent's forecasts.

Imagination, planning, and reasoning about the future are all tightly intertwined. They can be implemented more or less as just described and made available to the agent as actions. Policies can be formed that include these actions, such as "go to train station; find the next train to London; take the next train to London," eventually finding a train to London will probably itself involve a complex search and planning process. Because policies can include planning and imagination, forecasts can also, so the agent can forecast its chance of arriving successfully in London, despite not yet knowing exactly which train to take or even how exactly it will find a train. These forecasts can then be used for building higher-level knowledge, too. What if the agent wants to estimate the size of a room with many obstacles? (See Section 3, Layer 11). It could estimate the distances in two ways: by forecasting the number of steps needed to get from one side of the room to the other while avoiding the obstacles, or by imagining the appearance of the room without obstacles and using its normal forecasts to predict the time needed to reach the opposite side under those conditions. Such forecasts could be useful for characterizing spaces in which the obstacles regularly change.

Notice that imagination can modify forecasts without direct confirmation or verification in the sensorimotor stream. This is a general problem with any kind of imagination or planning in any system. It is therefore important to handle imagination with care.This is a known problem with planning an imagination, and some progress has already been made dealing with it [**?**].

## 4.5 Option models

To choose the best action in a given situation, a predictive agent may want to know all the likely consequences of taking that action or option. A one-step model of the environment estimates the next state given the current state and action: $\hat{s}_{t+1} = m(s_t, a)$. That model can then be used to predict the consequences of taking any action, and these predictions can be combined sequentially to predict longer-term consequences of action sequences. The obvious problem with these models is that they work well only for simple environments with well-chosen time scales. If the temporal resolution is too fine (i.e., the amount of time that elapses between one moment and the next is very small), then making even short-term predictions of the future quickly becomes intractable. If the temporal resolution is too coarse, then important details may become invisible. Forecasts avoid problems of temporal resolution by estimating the outcomes of *options* (whose duration is unspecified) and updating these estimates at every time step. Since the final outcome of the option does not depend on the temporal resolution, learning these estimates is stable and tractable.

Option models [27, 28] are similar to forecasts but predict the complete state vector that results from following an option to termination. Thus, an option model, $m_\mu$ estimates the future state based on the current state and a way of behaving as specified by an option, $\mu$:

$$\hat{s}' = m_\mu(s_t, \mu).$$

In other words, it predicts the value of each forecast. If the forecasts are chosen wisely, they are essentially orthogonal and independent, then the full distribution of future states is well represented by the resulting state vector. The option model allows planning at the level of options, which are of arbitrary duration. What we really want is a set of orthogonal, statistically independent basis functions $\phi(s)$ that compactly encodes the state, allowing estimation of the full set of forecasts, $\hat{F}$. The compact set could be used for predicting the results of the options: $\phi(s') = m_\mu(\phi(s_t), \mu_t)$. Any individual forecast $\hat{f}^i$ could then be extracted from the smaller set as needed, $\hat{f}^i(\phi(s))$.

## 4.6 Fast Learning and the Construction of new Forecasts

Sometimes it is necessary to learn quickly. You move your glass from the table to the counter. You then turn around and realize you are thirsty. Do you go to the table or to the counter? The glass had been on the table for quite a while, but you now know that it is on the counter because you learned—very quickly—when you moved it to the counter that it is no longer on the table. In the predictive agent, much of his learning is simply state update of forecasts that change in light of new information. However, we expect the predictive agent also to be able to learn new temporary relationships quickly. Learning quickly can be done quite easily with an MLP simply by increasing the learning rate, (though the cost, of course, is equally quick forgetting). We should expect that the agent can update its knowledge actively and modify simple forecasts from its experiences quickly. Exactly how this should be done is still under investigation and is beyond the scope of the current article. However, it does not appear to be an intractable problem. Learning should be quick for all aspects of knowledge maintenance, including knowledge construction, planning, and imagination. After the agent has imagined the result of going outside with a jacket on and determined it would be uncomfortably hot, it should be able to retain that forecast long enough to put the jacket back in the closet. We envision a process whereby new forecasts and new policies can be created on the fly to retain short-term knowledge with immediate relevance.

# 5 The Limits of Predictive Knowledge

As we have seen, predictive representations provide a powerful, isolaminar description of knowledge and state. In fact, we are cautiously hopeful they may eventually provide a framework for describing everything we currently think of as knowledge. But perhaps there are some essential weaknesses we have overlooked. Certainly, researchers from different backgrounds will look at this representation in different ways, asking whether it is powerful enough. This section attempts to expand the scope of that shown in the demonstration section and to discuss whether predictive representations are powerful enough to handle a broader spectrum of knowledge.

We first note that many basic building blocks of standard AI (e.g., symbols, objects, numbers, etc.) are in fact extremely sophisticated. Indeed, we humans are expert at forgetting how much effort we have put into our learning. Much of the knowledge we find so simple as to require no explanation (e.g., what we refer to so insouciantly as "object" and "number") is formed over the course of months or years of full-time processing by the world's most sophisticated cognitive computational device. Once a concept or skill is acquired and mastered, new challenges appear, and those months of frustration dissolve into history. What could be simpler than identifying and counting a few objects? The complex skills we picked up over decades of work now seem trivially easy. Yet, if they are so simple, what were we doing all those years as we struggled to acquire them? We were building up extremely rich networks of relationships, all founded in our interactions with the world. Thus, a thorough, or even partial treatment of this network of relationships is not feasible in any small space. Instead, we only hope to offer wisps of an outline, a sketch to provide insight and inspiration.

In each of the following subsections, a question is first raised about the possible limits of the representation, and then a suggestion is given describing possible scenarios based on predictive representations.

## 5.1 Object Permanence.

**Question**: Drescher showed that his schema mechanism, based on Piaget's schemata, could produce high-level abstract concepts, noting that "A constructivist system's greatest challenge is to transcend its initially supplied terms of representation, to extend its owns ontological vocabulary, to designate kinds of things that are radically different from any that it had previously been able to represent." His agent demonstrated the concept of object permanence by discovering that a (particular) visible object was the same thing as (the same) palpable object. What kind of understanding could a predictive agent develop of physical objects? When a predictive agent learns the patterns of interaction in its environment, to what extent is it developing an understanding of the object? Does it truly understand the concept of object permanence?

**Response**: This is an important question and allows the opportunity for many insights into predictive representations. Let us take two real-world objects as examples—a toy duck and a spoon—and let us consider what kind of an understanding of these two objects might develop in a predictive agent with a set of sensors and effectors similar to our own. (Note that Section 3 gave examples of particular patterns of interaction that could be achieved with very limited sensors and effectors. Such patterns were available, for example, when it interacted with doorways, walls, rooms, etc. With a more sophisticated sensorimotor apparatus, one that can lift, hold, turn, and weigh small everyday objects, it should be possible—but not necessarily *easy*—to describe much more sophisticated patterns of interaction with many kinds of objects.)

We give the agent the toy duck to manipulate with its hands. The agent builds forecasts that describe the toy duck's pattern of interaction, including the visual and tactile observations that result from its various manipulations: because of the toy duck's shape, the agent makes predictions about how what it feels and sees will change as it turns the duck this way and that; the duck's texture results in forecasts about the sensory stimuli that could be received from running a fingertip over it in different places and different ways; its variations in elasticity result in pressure differences when pressing in different positions, etc. Any important, discernable physical properties of the duck should be captured by these forecasts. Now we place the spoon in the agent's hand and the agent builds a set of forecasts describing the spoon's pattern of interaction. The agent can distinguish the toy duck and the spoon to the extent that they have different patterns of interaction. Each forecast is also a procedure that helps the agent verify whether it is holding the one or the other. But not all forecasts will be different for the two objects. If two objects are very similar, then their patterns of interaction will also be similar and the agent's forecasts will reflect that.

Now, if we put the toy duck and the spoon at a distance, the agent might still be able to see them, but would not immediately be able to distinguish them or to predict the pattern of interaction it could have with them. However, by repeatedly approaching and then interacting with them, the agent can learn to predict (forecast) the entire set of forecasts from the visual imagery alone. At that point, when the toy duck is visible, all forecasts pertaining to its pattern of interaction take on the values appropriate to the expected possibilities of interacting. As described in the appendix, these probabilities might all shift suddenly in recognition of the agent's opportunity to achieve the object's pattern of interaction.

If the toy duck or spoon is moved to a different place, the agent can search for it by having an option that moves its head and eyes more or less randomly and terminates when forecasts of the pattern of interaction suddenly become positive. This search policy can be improved through learning so that it is less random in familiar situations. Once found, state updates and fast-learning techniques can build policies that return the agent to states where these familiar patterns of interaction are available.

What other abilities might be implied by the notion of object permanence? The map technique from Section 3 can be used for any object's pattern of interaction so that its location relative to the agent can be maintained as the agent or object are moved. If the agent moves the toy duck, the agent can update its map to reflect its new relationship with the object. If an object is moved by some other agency and the agent can see it move, then based on what it sees, it can update its predictions of how to return to the object's pattern of interaction.

If the agent turns away and can no longer see an object, it can still maintain probabilities for the pattern of interaction it could obtain by turning back. These probabilities are learned and depend on the object and the environment. Similarly, if someone hides an object, say by placing a screen in front of the toy duck, the predictive agent can estimate the probability that it will be able to find the object through search, having learned this from its experience with the environment.

## 5.2 Complex Objects and Skills

**Question**: The demonstration section is quite simple in the sense that the robot is extremely limited, having very limited sensors and effectors, and the examples only showed how to build up certain specific kinds of knowledge directly related to those sensors and actuators. The big question is scaling: can this representation scale to complex sensorimotor systems like that of a humanoid robot? Can it handle complex objects, like pillows and keys, and complex skills, like walking, talking, and skiing? Or will it be unable to handle the vast complexity of interactions possible between complex sensorimotor systems and complex objects?

**Response**: These are critical questions, and the answer is: We do not know yet, but we have reason to be optimistic. This is why: After a bit of thought, it is not particularly difficult to break down individual examples of complex knowledge and skills, translating them each into simpler predictive statements about the sensorimotor stream. Each of these concepts also seem to lend themselves to being further broken down and broken down.[21]

Let us take a pillow as an example. How can an agent represent a pattern of interaction as complex as a pillow for a sensorimotor device as complex as a humanoid robot? It would consist of a set of forecasts about what the agent will experience when interacting with the pillow: the deformations the agent perceives when pushing on it from the sides, the feeling of it when placed under the head, the way it tucks under

[21] Of course, nearly the same can be said about symbolic knowledge representations, but with a very significant difference; we know now that very abstract forecasts can be built directly from the sensorimotor stream.

the chin while putting the pillow case around it, the way it squishes in the arms when it is squeezed, the amount of force you have to apply to your joints to hold it in front of you, and on and on. Each of these are predictive representations that capture part of what it is like to interact with a pillow. Each can be further broken down into other sets of predictions about visual and tactile observations and perceptions, and about tensions, forces, pressures, etc, between the hands and arms when holding it.

It soon becomes clear that the problem with real-world knowledge is not the problem of understanding things in predictive terms, but of dealing with the *quantity* of forecasts that need to be resolved. The demonstration of Section 3 is misleading in one important respect: it reaches high levels of abstraction in just twelve layers. But that demonstration is intentionally broad to show the potential of the representation. It may make more sense to think instead of hundreds, maybe thousands of layers, or maybe even of thousands of layers per day. The continual-learning process might build up millions of levels, one layer at a time, in the same repeating pattern: identifying important new perceptions, setting them as targets, and learning to control them. To learn a new skill, we first need to recognize that we have achieved it, then we can practice getting there. Learning to stand, for example, is a layered process of first recognizing when one is standing, then learning how to get into that situation. To put on your shoes you might first want to line them up side by side, but learning to line them up requires first learning to recognize their proper positioning (when properly positioned, the tips curve in toward the middle), then learning how to get them into that position. Grasping an object requires first recognizing that an object has been grasped, and then learning a policy to achieve that state.

After the policies have been learned, forecasts can be made about whether one can stand, grasp, or put on one's shoes from the current situation. These forecasts can then be used to describe new achievable targets. To teach my son how to put a book into the bookshelf, I first had to teach him to distinguish the binding from the other sides. Only after he could successfully make this distinction could I ask him whether the binding was showing. Then after he could answer that question, I could ask him to put the book back so that the binding was showing. And this process probably happens dozens of times a day.

Every time the world responds to us in a way we were not expecting, we have discovered something interesting to forecast. We can then build a policy to maximize or minimize that forecast. Unlike large reinforcement learning tasks, where rewards can be quite distal and policies can be quite elaborate, knowledge construction with GPK can focus learning on temporally proximal targets. Option policies can be built on top of existing KUs to maximize or minimize a relatively immediate forecast quantity.

Skills increase in complexity a little bit at a time. We look at complex skills and wonder at their intricacy, but we should keep in mind the amount of time, the years and years of focused learning that were required to master those skills.[22] From a GPK perspective, the skills are formed from an enormous number of forecasts and options. But the individual layers are simple, targeted forecasts. Learning to walk entails (a) forecasting the many different ways that one can stumble and then (b) learning to avoid them; each a different forecast, each a different skill, each built in thin layers upon the previous ones. We visualize these layers less like the stories of a building and more like the thin layers of an onion, or perhaps the fine coatings of a pearl.

Because policies are learned independently, computation can be distributed across many processors, as resources permit. It may be possible to improve resource efficiency and ameliorate the curse of dimensionality by projecting the options and forecasts onto a two- or three-dimensional parallel computational platform, such that nearby options and forecasts are similar and depend on similar inputs [21]. Finally, forecasts can be learned using MLPs with many layers, and there is now good evidence that deeply layered MLPs are capable of performance better than that of humans—even in recognition tasks where humans have (until recently) always been superior [**?**].

### 5.3 Number and Math

**Question**: Are predictive representations limited to abstract knowledge of the physical world, or can it also be used for representing mathematical knowledge?

[22]To get a good idea of how much training goes into a skill you now take for granted, try writing several pages with your non-dominant hand. Improvement does not come quickly or easily.

**Response**: Perhaps mathematical regularities seem to be deeper than physical ones (the same mathematical laws might apply in universes with different physical laws from our own), but we still have to learn them through experience with the physical world. This does not imply that mathematical truth is a construction of the mind, any more than Mt. Everest is a construction of the mind just because we have to learn about it. After years of hard training, some of us come to believe that basic addition is the epitome of cognitive simplicity, yet the simplicity is a terrific illusion that reflects perhaps how good we are at forgetting the effort it takes to build up our complex skills once the hard work has been done. I was struck when teaching my son to do addition that I had to start at a much lower level than I would ever have guessed. I first had to teach him to count, a rather difficult process that took some mastering, including learning the sequence of numbers, learning to separate the objects that had to be counted (in his case, toy cars) and learning not to count the same ones twice. Then, after much, much practice, there was another strange idea he had to make sense of: if you take a bunch of things, count them, rearrange them, and count them again, you always arrive at the same number. He was delighted to see this occur over and over until finally it was no longer a novel trick of the universe but rather a predictable event. Until then, he believed he lived in a universe in which the number of objects could change whenever they were rearranged. Just as we learn to expect what happens when we handle a familiar object, we learn to expect that the number of objects stays the same when they are rearranged.

Maybe we have to learn to make predictions about other mathematical events too: what will happen when we carry through one mathematical operation or another? We learn to predict how many objects there will be, or where a finger will end up on the number line, or what number will be written in a specific place. We use familiar skills and knowledge of everyday relationships to help us learn abstractions about hard-to-comprehend mathematical truths. We envision surfaces that stretch or pieces of string that intertwine. This is difficult mental work, requiring focused imagination, but over time we learn to forecast the eventual results of these imaginings, possibly in the same way that we learn to forecast the results of physical interactions, giving rise to many learned layers of abstraction. In other words, we imagine, we plan, and we take short cuts. And so it would make sense that we should be faced from time to time (or maybe many times every day) with mathematical truths that defy our expectations, because our intuitions are forecasts based on our experiences and imagination, and those experiences do not necessarily lead to perfect estimates.

Perhaps it is not inconceivable then, that mathematical knowledge can also be based on predictions of essentially the same sort as those we make about physical regularities. What sets them apart is their dependability and generality. Once we adequately learn mathematical predictions, they seem to be extremely dependable and general, equally true regardless of when or where they are made, or about what. And this makes them very special.

## 5.4 Variable Binding and Logical Quantifiers

**Question**: Traditional symbolic representations offer a great deal of power, allowing simple, dense and highly general specifications of knowledge. They can, for example, easily deal with type-token distinctions (recognizing the difference between a thing and its type; for example, distinguishing an actual physical chair and the type *chair*), the binding problem (recognizing the properties of physical objects; for example, recognizing that a red triangle next to a green square is an object with the two properties *triangularity* and *redness*), and logical quantification (encoding the fact that all things or some things of a certain type have a certain property; for example, encoding the existence of a white swan, or that all swans are white). These issues have generally been difficult for sub-symbolic systems [**?**], and in many ways humans seem to behave more like symbolic systems. Do forecasts somehow avoid these traditional problems with sub-symbolic systems?

**Response**: Let's look at these individually. As for type-token distinctions: what is the difference between a type and a thing itself? There is still considerable debate among practitioners about what constitutes a type in subtle cases. Part of the power of predictive representations is that, in connecting all abstractions to explicit statements about the sensorimotor stream, it avoids certain areas of confusion. Perhaps the type-token distinction reflects some confusion about the nature of generalization, and perhaps what we really want is an agent that can interact successfully with things in a robust way and generalize its abilities meaningfully. Some may debate the ways in which the Platonic type "chair" is different from any individual chair, but what the sensorimotor stream provides is the opportunity to sit down on some things and not on others. We are satisfied to have a way of describing sitting (as a policy) and to forecast whether that policy can be completed successfully given what is currently known about the environment. We only note that this policy will be different for every individual and so will the forecasts about it.

The binding problem is more concrete. Here is a scenario whereby a predictive agent might be said "to know" that an object is both red and square: a predictive agent might forecast that it will observe red if it follows a certain option to termination. That option might entail moving its eye(s) to a particular place. Another option forecasts that it will see a square shape if it follows *the same* option. Together, the forecasts predict the presence of both perceptions, and the option describes how they can be obtained. This scenario can be generalized as follows. Let us imagine that there is an actual object somewhere in the agent's environment that is both red and square. The agent can keep track of that object in the same ways that were described above. It can have a policy that will navigate to that object. It can update its forecasts as it or the object changes location. Now the agent knows how to get to a state where it will observe redness and squareness. If the MLP cannot easily learn to distinguish a square from, say, a triangle, then to the extent that it is important to distinguish them, the agent can have an option whose policy terminates differently for each (perhaps by counting the sides or angles or simply by examining the sides or angles more closely), and the forecasts for that option could then be used to make the desired distinction.

Even more powerfully, the agent may have an option that searches for the object until it is found; forecasts can be made for that option, in effect predicting whether it can find something whose pattern of interaction allows both the perception of red and the determination of squareness.

Now let us look at quantifiers. The following offers a scenario whereby a predictive agent might be said to understand existential and universal quantification. The line of thought is quite similar to that just shown. What might it mean to say that a predictive agent knows that something exists? A reasonable approximation might be that the agent knows it can find that thing and interact with it, meaning it can find a state where a certain pattern of interaction is available; i.e., it has an option that terminates when the pattern of interaction becomes available and the forecast for that option is high. If the agent has an option that will reliably terminate in a state where it can interact with a swan and observe white, and if it has a forecast predicting a high success rate for that option, then it does not seem unreasonable to say the agent then knows that a white swan exists.

What could it mean to say that a predictive agent knows that all swans are white? If the agent has an option that reliably terminates whenever it sees a swan, one forecast function $f_{\text{swan}}$ based on this option that predicts whether the option will terminate successfully, and another forecast function $f_{\text{white swan}}$ based on the same option where $c = 0$ and $z = \textsc{observe white}$, and if these two forecasts always have the same value, then one might reasonably assert that the agent knows that all swans are white. It is unclear how useful such forecasts would be to a predictive agent, but it is comforting to know that such predictions can indeed be represented.

## 5.5 Communication and Language

**Question**: Language seems in many ways to be primarily symbolic, yet predictive representations are not. Is there hope that predictive representations will also be appropriate for representing language?

**Response**: Let us look at a possible scenario as to how language abilities might begin in a predictive agent. Before learning to produce language on its own, a predictive agent will already have started to predict important events in the environment such as being comforted or fed. Attempting to forecast these events, it will notice sounds that it often hears preceding them, such as "here I come" or "bottle." The agent's forecasts for the important events will become informed by these *cueing* sounds, such that when the sounds are perceived, the forecasts will change. Meanwhile, the agent is trying to learn policies of its own to maximize the forecasts of things that it likes. The agent hears its own noises, and if they resemble a cueing sound, the forecasts for certain events will suddenly change, which might lead the agent to build a forecast for the cueing sound itself. It now finds that it can learn a policy to control that forecast with high reliability, thus leading to early babbling. Of course, the sound the agent makes might not resemble the cueing sound to our ears, but the agent's estimator for the forecast is still improving and its own rough production of the sound may have a noticeable effect on the estimator. As forecasts are constructed in layers, policies for the production of sounds are learned that try to maximize the forecasts as recognized by the estimator.

Similarly, any sounds (words) that frequently precede any other forecasted experience (such as recognition of objects or activities) become incorporated into the forecast for that experience, so reproducing such sounds will have an impact on those forecasts. Of course, there are untold numbers of social ramifications and consequences of speech. The social effects of one's speech are observable and can be (at least mildly) forecasted; the resulting forecasts might draw on the agent's knowledge (forecasts) about the listener. Thus, the agent also begins to forecast the social consequences of its own speech. (See more in Section 5.8.)

It is a long journey from there to sentence construction, and there are no doubt many steps along the path, but perhaps this scenario could prove a useful beginning.

## 5.6 Historical Events

**Question**: If predictive representations contain only predictions about the future, how can a predictive agent retain information about historical events?

**Response**: An agent that has only predictive representations might show no deficits with respect to knowledge of historical facts. I am always a bit astonished by my (very bright) four-year-old son's inability to recount historical events. He certainly knows quite a bit about the past and can answer many specific questions (e.g., "what kind of car did we have when we lived in Switzerland?" or "which park did you go to today?" etc.) But he cannot yet produce anything like a narrative about most events that occur during the day (despite the incentives I give him to do so). Thus, it could be that for some people, such an ability takes a long time to develop. An artificial agent whose knowledge is stored solely as forecasts might have a similar developmental process because it only needs to predict what will happen in the future.

But at some point knowledge of specific historical events may become useful for predicting the future. Perhaps only after the development of language and social interaction, when one starts to anticipate the questions that one will be expected to answer. At that point, the agent might be able to repurpose its predictive mechanisms to anticipate questions about events of the past (by creating forecasts) and to build policies that answer those questions. Thus, perhaps knowledge of the past can be captured as a series of predictions about conversations one could have, books one could read, and resources one could appeal to for verification.

At first, this explanation may seem unsatisfying. We may think that we know facts about the past, what occurred in ancient Rome for example, even though we clearly do not have an internal historical record of the events from that time. But if in a certain situation an agent can forecast that it will see a wall if it turns right, this forecast can also be interpreted as a record of what it has seen in that situation in the past. In the same way, our predictions about what we would observe should we consult a history book or wikipedia on a certain historical subject should not be confused with an internal storage of history. We use the internal machinery we have to help us live in a world where historical information is important, just as we use it to cope with the world's physical regularities. And it does not seem implausible that a predictive agent's internal predictive machinery could allow it to make statements regarding the past.

## 5.7 Emotions

**Question**: Our feelings and emotions seem to be an integral part of our state. If state is nothing but knowledge, and knowledge is nothing but predictions, then this implies that there is some connection between emotion and prediction. Can emotions be represented predictively?

**Response**: This may be one place where predictive representations have little to say about human state or knowledge. However, a few things should be said. First, it may be beneficial to the agent to model the emotions of others, as emotional state (mood) is an important social indicator in humans. These models would consist of forecasts about the behavior of others (see 5.8). Second, to some limited extent, certain rough parallels could be drawn between human emotions and the predictive agent's forecasts.

The agent's value function $\hat{V}$ estimates the expectation of future reward. It may be a stretch, but probably not an entirely unreasonable one, to draw a parallel here with humans and say that an increase in value in the agent is something akin to a positive emotional response in a human, and a decrease in value is something like a negative emotional response. Could there also then be correlates to other, more specific emotional responses, like fear, hope, anger, and love? These are slippery questions; perhaps a more meaningful way to answer them is to ask instead what emotions a human might experience who made predictions similar to those of the agent. What would the human emotional response be to a general forecast of danger, or the possibility of great improvement in $\hat{V}$ (as described in Section 4.3)? How do humans respond to patterns in their environment whose appearance consistently results in an increase or decrease in $\hat{V}$? Along these lines, one might be able to trace out more intricate scenarios describing specific human emotional responses to specific kinds of predictions about the future; those predictions could then possibly be restated in terms of forecasts. This weak suggestion may be all that predictive representations have to say on this subject.

Finally, humans seem to have emotional responses to purely imagined scenarios. Though it may say nothing about emotional state, it also makes sense for the predictive agent to use its same machinery, including its value function estimator $\hat{V}(s)$ and all its forecasts $\hat{F}(s)$, to evaluate imagined situations, $s$.

## 5.8 People and Social Relationships

**Question**: Could predictive representations extend to social interaction? Could they capture knowledge about people's behavior, beliefs, preferences, desires, goals, awareness, consciousness?

**Response**: It is surprisingly easy to think of each of these in a predictive framework. We understand people just as we understand everything in our environment: as regularities in our sensorimotor stream that allow certain kinds of interaction. But our interaction with them is special because they are both very important and highly complex. Our knowledge of other people often reflects our interest in predicting what they will do. It is convenient to think of them as having goals (i.e., having policies that maximize or minimize a forecasted value), beliefs (forecasts), abilities (options) and preferences (a reward function that drives their behavior). We can make predictions about how other people will respond to our actions, which might entail predicting what *their* predictions (beliefs) are and how these might affect their behavior.

Just as we learn to predict complex properties of the environment that most impact us, we also learn to predict the (probably much more) complex properties of the people we interact with, and we try to foresee how they will react to us. Early in life one must discern things about people: is this person going to help me or not (i.e., in the presence of this person, do my predictions get better or worse for the things I want to achieve)? One learns to recognize signs that change one's predictions, e.g., smiles and frowns, attentiveness, a quiet or loud voice, etc. Tiny changes in these signals can lead to radically different futures, and therefore it is important to learn to predict and influence them. For example, if your boss is frowning, you should probably reconsider demanding a raise, because you might not keep your job if you do.[23] Predictions based on these signs better estimate how one's responses will be received. These are our predictions about other people's beliefs. If I predict that Bob will smile soon, I might say I think he's happy. If I predict that Bob will run out of the room, I might say I think he's afraid. I may make predictions about what things Bob might do or say when I interact with him in certain ways or tell him certain things, and I might describe these as my beliefs about Bob's beliefs. It seems that predictions such as these can also be layered: I might make predictions about Bob's beliefs about me or my beliefs (and so on), and I might find it useful to try to influence these.

The predictions in the previous paragraph could be expressed fairly naturally as forecasts, and though an actual implementation in an artificial agent would require a foundation of great sophistication, the representation itself does not seem inappropriate for capturing this *kind* of knowledge.

[23] Here hope and fear are mixed together in close combination, and choosing the right policy is critical.

In fact, a predictive agent may be able to describe many complex social phenomena (at least at a superficial level) fairly straightforwardly, such as: empathy (a forecast that interacting with a person will reduce a recent decrease in the agent's value-function estimate), helpfulness (a forecast that interacting with a person will accelerate the achievement of the agent's goals), cruelty (a forecast that interacting with a person will have a high probability of decreasing the agent's value-function estimate), maybe even love (a forecast that the agent's value function is significantly higher in those states in which interaction with the loved one is readily available). Naturally, much more subtlety could be added to each of these proposals. Furthermore, in each case, the agent could make forecasts about how humans (or agents) interact with each other by imagining itself in their position and forecasting the agent's own responses. Perhaps the agent could understand concepts as abstract as "fairness" and "justice" by imagining itself in the place of both parties to an agreement and forecasting how the agreement would affect its own value function.

Unexpectedly, predictions may have something helpful to say on the topic of consciousness. While little firm scientific progress has been made on this topic, one thing we can say uncontroversially is that people talk about it. To the extent that there is an *it* there to talk about, there must be an explanation for *why* we talk about it. Graziano and Kastner [5] propose that the root cause of the discourse is a neurophysiological mechanism adapted to social interaction; ultimately this mechanism provides a means of modeling and locating the cause of one's own activities and ascribing to that cause the same complexity of interaction we ascribe to others. This utterly pragmatic description is highly compatible with the mechanisms offered by predictive representations. People are mysterious: we devote enormous resources into predicting their behavior but are only mildly successful, though we continue to improve as we devote more time and effort. Our own behavior is at times equally mysterious because it results from a vastly complex network of causes and consequences. Once we begin to develop forecasts for predicting the behavior of other people, we can direct these same forecasts to predicting our own. The emotional responses we get from other people's subtle signs of mood can become focused on ourselves. We talk about the mystery of consciousness precisely because we cannot say what it is, though we attribute *it* to ourselves. We notice that we are *aware* of so many things: the color of the grass, the touch of the wind, the comprehension of one's place in one's surroundings. But there is no reason an artificial predictive agent cannot be aware of all these things: these forecasts and sensory observations. Ultimately, a predictive agent will become aware of its own behavior, will try to predict it, and will find it mysterious.

# 6 Conclusions

This paper began with the claim that one of the great problems in AI and Cognitive Science is that of connecting a representation to what it represents. So it is reasonable to ask, how do predictive representations deal with this problem? They deal with it by simply moving out of its way. We suggest this problem may be an artifact of a misplaced expectation: the belief that there can be knowledge without interaction. To us it is intuitive and clear that knowledge is fundamentally about interaction, because all knowledge must be verifiable in the sensorimotor stream [24]. The agent can only know things about its environment by interacting with it, by building up an awareness of how the things it interacts with might respond.

For decades, constructivists in AI and Cognitive Science have sought a simple mechanism that allows isolaminar construction of knowledge and skill. Predictive representations offer a new alternative. The power of predictive representations is that they describe knowledge and state in terms ultimately most important to an autonomous agent: as predictions about the future. In particular, *forecasts* predict what the agent will perceive as a result of how it behaves.

Forecasts are scalar values and can be layered in multiple ways: they can be used as inputs by other forecasts ($\hat{f}$), they can serve as the targets for other forecast functions ($f$), and they can serve as the initiation condition ($I$) and termination condition ($\beta$) of the agent's options. As a result, they allow formation of meaningful abstract perceptions based exclusively on the agent's sensorimotor stream.

Forecasts have qualities that fulfill many of an autonomous agent's representational needs: they make predictions about the future conditioned on arbitrary ways of behaving and over arbitrary time spans; the timeframe of a prediction can be described very precisely or very generally; they are more flexible, more intuitive, and more general than one-step predictions; they suffer no extra computational burden at high temporal resolution; they can represent very abstract knowledge in a way that is fully verifiable in the sensorimotor stream; they can be composed and layered in multiple different ways; they can be learned simultaneously from all applicable data; they use known learning methods guaranteed to converge; and they can be distributed across multiple parallel processing devices.

One of the more interesting aspects of predictive representations is that they have a separate recognition and production system, where the recognition is embodied by the MLP and the production is embodied by the forecast functions and learned options. In many ways this seems to parallel the differences in our own abilities to recognize and generate. Many things that we can recognize easily we may find extremely difficult to produce ourselves; for example, imitating an other person's voice or accent, or drawing an object or a face. We might even be expert at recognizing subtle differences in the quality of things that we absolutely cannot reproduce ourselves (e.g., music, poetry, art, wine) or even adequately describe. Such parallels between human cognition and predictive representations, which were shown throughout the paper, are alluring, though they have yet to be investigated experimentally. However, it does not seem implausible that we too describe our world in largely predictive ways and that our knowledge, our highest-level perceptions and abstractions, are in fact merely compositions of predictions about how our actions might affect our future. One obvious but profound logical consequence of describing knowledge as action-conditional predictions about the future is this: state (i.e, one's full understanding of one's environment), is thus equivalent to what one can predict about the future, which is equivalent to all of one's knowledge.

Unlike many previous constructivist authors, we have not provided a method of construction: we have not described when or how new forecasts and options should be added. Such a method is clearly required for a fully autonomous agent. However, just as an architect must understand the materials available before beginning a new design, it is important to first understand and have confidence in a representation before beginning development of a method of construction.

Whether or not forecasts will be an effective mechanism for bridging the gap between the sensorimotor stream and abstract human-level knowledge remains to be proven, but this paper has offered a strong argument that they will eventually be able to do so. We have attempted to provide a thorough and general treatment of their properties, but most importantly have given a detailed demonstration of their power, showing how they allow isolaminar construction of useful abstractions at every level. Though the hypothetical agent of the demonstration (Section 3) had extremely limited sensors and effectors, it could represent knowledge about increasingly abstract aspects of its world in each of twelve layers. At every level the agent made predictions (forecasts) about what it would perceive (observe or forecast) after behaving in certain ways (following certain options).

We have also considered many seemingly complex representational issues, from object permanence to mathematical knowledge to language and social interaction, and we have discussed how they could potentially be addressed by predictive representations, often in novel ways. The sketches in Section 5 above were meant to give a glimpse of their potential rather than to claim a definitive solution. We hope that they reveal something about how predictive representations might be extended through vast numbers of layers of abstraction to address highly complex kinds of knowledge, and though these subjects may sometimes involve great complexity, it does not seem implausible that predictive representations could address them in useful ways.

There are still many challenges for the future and this paper provides only a forecast that this future is worth pursuing. We have been careful to claim only that some useful and surprisingly sophisticated concepts can be clearly and easily described with predictive representations. We have not proven that *all* knowledge can be described in this way, but we also do not rule it out. Perhaps we *will* encounter some important kinds of concepts that cannot adequately be described using predictive representations; but would that mean the predictive representations are inadequate, or that the concepts are ill-formed? It is not yet clear at this moment.

We in the AI community are accustomed to clear definitions, firm divisions, stark boundaries. But knowledge may not actually be that way. Perhaps it is only an illusion that our knowledge is clear and well-defined. Perhaps knowledge is really much messier, much slipperier, much more subjective, having many vague relationships and boundaries. If knowledge is the former, then we should eventually be able to define it all, to pin it down once and for all, to know for all time what makes a bird a bird, what makes a pillow soft, what makes a game delightful. But if we cannot expect our concepts to be satisfyingly concrete, succinct, perfect, and universal, then we must accept the possibility that our representations of knowledge will be as unbridled as the knowledge itself: as vague, as messy, as subjective, and as free.

# 7 Acknowledgments

# 8 Appendix

## .1 Locking in

{The appendix is work in progress.} When a predictive agent receives information that should support its current predictions, how is that information used to bolster its current estimates? It searches for a stapler in a room and then the stapler gradually comes into view. At first there is minor evidence that might indicate the presence of the stapler, then more and more, but how is new evidence combined with the evidence seen so far such that the forecast is strengthened, is resilient in the face of noise, and continues when the agent looks away? We call this the "locking in" problem because the evidence should accumulate until the agent is convinced that the stapler is present, at which point, the forecasts regarding the stapler's pattern of interaction are "locked in," and the agent knows that specific ways of interacting have become available.

In the simplest case, the robot has an action (say *ff* from Section 3.1), that results in an observation (say the signal from the TOUCH sensor, $o \in \{0, 1\}$), and the robot can be in one of two situations: in situation $A$ the robot will touch something if it moves its finger forward, and in situation $B$ it will not. The sensor is noisy, so if we place the robot facing something that it can touch (e.g., a wall) and it moves its finger forward, there is a probability $P_A$ that it will observe a 1, and if there is nothing there to touch, then there is a different probability $P_B$ that it will observe a 1. How does the agent *decide* at any moment whether it is in state A, where it can reach out and touch something, or state B, where it cannot? That is, how do its forecasts lock in on the prediction that if it again reaches its finger forward it will (or will not) observe a 1 through its TOUCH sensor? The forecast is over an option $\{\pi = \mathit{ff}, I = 1, \beta = 1\}$ whose policy is a single action *ff*, which can be taken in any state and which always terminates immediately. The target is $\{c = 0, z = \mathit{touch}\}$, which is the value of the touch sensor. Therefore, the forecast should predict $P(\mathit{touch} = 1 \mid \mathit{ff}) \equiv P(1 \mid a)$, where, for convenience and generality, we use 1 as an abbreviation for $\mathit{touch} = 1$ and $a$ as the name of the action or option. But these probabilities should also be contingent on the history of observations in response to this action. Let $x$ be the most recent sensor observation in response to $a$, and let $h$ be the history of all previous sensor observations so far in response to this action, so that $\hat{f}(s_t) = P(\mathit{touch} = 1 \mid \mathit{ff}, h) \equiv P(1 \mid h)$, since we are only concerned with that part of the history where the action *ff* was taken. Then,

$$\begin{aligned}
\hat{f}(s_t) = P(1|h) &= P(1|A)P(A|h) + P(1|B)P(B|h) \\
P(A|hx) &= \frac{P(hx|A)P_A}{P(hx)} \\
&= \frac{P(h|A)P(x|A)P_A}{P(h|A)P(x|A) + P(h|B)P(x|B)P_B} \\
&= \frac{\frac{P(A|h)P(h)}{P_A}P(x|A)P_A}{\frac{P(A|h)P(h)}{P_A}P(x|A)P_A + \frac{P(B|h)P(h)}{P_B}P(x|B)P_B} \\
&= \frac{P(A|h)P(x|A)}{P(A|h)P(x|A) + P(B|h)P(x|B)}.
\end{aligned}$$

If the state vector consists of the current observation $x_t$ and the current forecast $\hat{f}(\mathbf{s}_{t-1})$, then this can be rewritten as:

$$\hat{f}(\mathbf{s}_{t+1}) = \frac{P(A|h)P(x|A)}{P(A|h)P(x|A) + P(B|h)P(x|B)}.$$

{I think it takes a funky 3-layer MLP to calculate this.}

$$\begin{aligned}
P(1|h) &= P(A|h)P_A + P(B|h)P_B \\
P(1|h) &= P(A|h)w_A + P(B|h)w_B \\
P(A|ho) &= \frac{P(ho|A)w_A}{P(ho)} \\
&= \frac{P(h|A)P(o|A)w_A}{P(h|A)P(o|A) + P(h|B)P(o|B)w_B} \\
&= \frac{\frac{P(A|h)P(h)}{w_A}P(o|A)w_A}{\frac{P(A|h)P(h)}{w_A}P(o|A)w_A + \frac{P(B|h)P(h)}{w_B}P(o|B)w_B} \\
&= \frac{P(A|h)P(o|A)}{P(A|h)P(o|A) + P(B|h)P(o|B)}
\end{aligned}$$

And similarly for $P(B|ho)$. Figure ?? shows a sample interaction for the value of $P(1|h)$ when the agent is in situation $A$ and $P_A = 0.9$, and also when $P_A = 0.8$.

The picture is more complex than this, however, in that there is always some chance that the robot's situation might change and then its forecast must also change. Say the probability of switching from situation A to situation B is P(switch), then

$$\begin{aligned}
P(A|ho) =&(1 - P(switch)) * \frac{P(A|h)P(o|A)}{P(A|h)P(o|A) + P(B|h)P(o|B)} \\
&+ P(switch) * \frac{P(B|h)P(o|A)}{P(B|h)P(o|A) + P(A|h)P(o|B)}
\end{aligned}$$

Because there is some chance that no switch has occurred (first line), in which case the calculation would be as in the simple case, and there is some chance that a switch has occurred (second line), in which case the previous $P(B|h)$ and $P(A|h)$ values were reversed, so should be handled in reverse. The overall update just is the first case, multiplied by its chance of occurrence plus the second case, multiplied by its chance of occurrence.

In the same way, any perception becomes substantiated by evidence. Consistent pieces of evidence will be mutually reinforcing. Figure ?? shows an interaction in which the robot changes from situation A to situation B, with the same probabilities far $P_A$ as in the previous graph.

# A References

# References


[1] Gianluca Baldassarre, Marco Mirolli, Francesco Mannella, Daniele Caligiore, Elisabetta Visalberghi, Francesco Natale, Valentina Truppa, Gloria Sabbatini, Eugenio Guglielmelli, Flavio Keller, Domenico Campolo, Peter Redgrave, Kevin Gurney, Tom Stafford, Jochen Triesch, Cornelius Weber, Constantin Rothkopf, Ulrich Nehmzow, Mia Siddique, Mark Lee, Martin Huelse, Juergen Schmidhuber, Faustino Gomez, Alexander Foester, Julian Togelius, and Andrew Barto. The im-clever project: Intrinsically motivated cumulative learning versatile robots. In L. Canamero, Y.-Y. Oudeyer, and C. Balkenius, editors, *Proceedings of the Ninth International Conference on Epigenetic Robotics*, pages 189–190, Lund: Lund University, 2009. Lund University Cognitive Studies.

[2] Joseph D. Becker. A model for the encoding of experiential information. In Roger C. Schank and Kenneth Mark Colby, editors, *Computer models of thought and language*, chapter 10, pages 396–434. W. H. Freeman and Company, 1973.

[3] Michael Cunningham. *Intelligence: its organization and development.* Academic Press, 1972.

[4] Gary L. Drescher. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence.* MIT Press, Cambridge, Massachusetts, 1991.

[5] Michael S. A. Graziano and Sabine Kastner. Human consciousness and its relationship to social neuroscience: A novel hypothesis. *Cognitive Neuroscience*, 2(2):98–113, 2011.

[6] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory.* John Wiley & Sons, New York, 1949.

[7] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, Michigan, 1975.

[8] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2004.

[9] Herbert Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6):1371–1398, 2000.

[10] William James. *The Principles of Psychology.* Dover Publications, June 1950. Originally published 1890.

[11] Benjamin J. Kuipers. The map-learning critter. Technical Report TR85-33, University of Texas, Austin, 1985.

[12] Benjamin J. Kuipers and Yung-Tai Byun. A robust, qualitative method for robot spatial learning. In *The Seventh National Conference on Artificial Intelligence*, 1988.

[13] Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[14] David Pierce and Benjamin Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–229, 1997.

[15] Eddie J. Rafols. Temporal abstraction in temporal-difference networks. Master's thesis, University of Alberta, 2006.

[16] Eddie J. Rafols, Mark B. Ring, Richard S. Sutton, and Brian Tanner. Using predictive representations to improve generalization in reinforcement learning. In L. P. Kaelbling and A. Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 835–840, 2005.

[17] Mark B. Ring. Incremental development of complex behaviors through automatic construction of sensory-motor hierarchies. In Lawrence A. Birnbaum and Gregg C. Collins, editors, *Machine Learning: Proceedings of the Eighth International Workshop (ML91)*, pages 343–347. Morgan Kaufmann Publishers, June 1991.

[18] Mark B. Ring. Two methods for hierarchy learning in reinforcement environments. In J. A. Meyer, H. Roitblat, and S. Wilson, editors, *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 148–155. MIT Press, 1993.

[19] Mark B. Ring. *Continual Learning in Reinforcement Environments.* PhD thesis, University of Texas at Austin, Austin, Texas 78712, August 1994.

[20] Mark B. Ring. CHILD: A first step towards continual learning. *Machine Learning*, 28:77–104, 1997.

[21] Mark B Ring, Tom Schaul, and Jürgen Schmidhuber. The Two-Dimensional Organization of Behavior. In *First Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics*, Frankfurt, 2011.

[22] Jürgen Schmidhuber. On learning how to learn learning strategies. Technical Report FKI–198–94 (revised), Technische Universität München, Institut für Informatik, January 1995.

[23] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[24] Richard S. Sutton. Verification, the key to ai. http://rlai.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html, 2001. Available online.

[25] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[26] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 761–768, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[27] Richard S. Sutton, Doina Precup, and Satinder P. Singh. Intra-Option Learning about Temporally Abstract Actions. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 556–564, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[28] Richard S. Sutton, Doina Precup, and Satinder P. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[29] Richard S. Sutton, Eddie J. Rafols, and Anna Koop. Temporal abstraction in temporal-difference networks. In *Advances in Neural Information Processing Systems 18 (NIPS*05)*, pages 1313–1320. MIT Press, 2006.

[30] Richard S. Sutton and Brian Tanner. Temporal-difference networks. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.