# Intrinsic Motivation For Reinforcement Learning Systems

Andrew G. Barto and Özgür Şimşek
Department of Computer Science,
University of Massachusetts Amherst
barto@cs.umass.edu ozgur@cs.umass.edu

*Abstract*— **Motivation is a key factor in human learning. We learn best when we are highly motivated to learn. Psychologists distinguish between extrinsically-motivated behavior, which is behavior undertaken to achieve some externally supplied reward, such as a prize, a high grade, or a high-paying job, and intrinsically-motivated behavior, which is behavior done for its own sake. Is there an analogous distinction for machine learning systems? Can we say of a machine learning system that it is motivated to learn, and if so, can it be meaningful to distinguish between extrinsic and intrinsic motivation? In this paper, we argue that the answer to both questions is "yes," and we describe some computational experiments that explore these ideas within the framework of computational reinforcement learning. In particular, we describe an approach by which artificial agents can learn hierarchies of reusable skills through a computational analog of intrinsic motivation.**

## I. INTRODUCTION

The concept of motivation refers to the forces that energize an organism to act and that direct its activity. Psychologists distinguish between *extrinsic motivation*, which means being moved to do something because of some specific rewarding outcome, and *intrinsic motivation*, which refers to being moved to do something because it is inherently enjoyable. Intrinsic motivation leads organisms to engage in exploration, play, and other behavior driven by curiosity in the absence of explicit reward. In this paper we consider what it means to make an artificial learning system, specifically an artificial reinforcement learning (RL) system, intrinsically motivated.

The idea of designing forms of intrinsic motivation into artificial learning systems is not new, having appeared, for example, in Lenat's AM system [11] and within the framework of computational RL in work by Schmidhuber [15] and others. Space does not permit a thorough acknowledge of all the relevant previous research. Our efforts on this topic began recently when we realized that some new developments in computational RL could be used to make intrinsically-motivated behavior a key factor in producing more capable RL systems. This approach, introduced by Barto et al. [1] and Singh et al. [17], combines intrinsic motivation with the notion of an "option" as defined by Sutton et al. [20]. In this paper, we describe this approach and present an alternative to the algorithm given in refs. [1] and [17].

## II. BACKGROUND

### A. RL and Motivation

The psychologist's concept of motivation is not usually associated with machine learning, but there are parallels between the motivated behavior of animals and the behavior of an RL system as it "tries" to maximize reward. This is strengthened by the striking correspondence between important components of RL algorithms and the activity of dopamine neurons [16], which play an important, though incompletely understood, role in animal motivational systems. While a wide gulf still separates animal motivational systems and computational RL algorithms [4], a good case can be made that the gradient of an RL system's value function (which is basically the same as the temporal difference, or TD, error [18]) acts very much like "incentive salience" in directing behavior [12]. If we said that *an RL system's behavior is motivated by the gradient of its evaluation function*, we would be consistent with contemporary theories of animal motivation.

Part of what an RL system learns, at least one that uses a value function, is to make sure its value function provides accurate estimates of expected future rewards. Consequently, tying down the entire behavioral and learning processes is a reward function: a real-valued function of the decision problem's states and actions. It is given as part of definition of the learning problem that the system is faced with solving. The RL system's objective is established by the given reward function, without which the learning problem to be solved would not have a coherent definition. This would argue that an RL system is clearly extrinsically motivated because it works to achieve externally supplied rewards.

Is it meaningful for an RL system to define its own internal rewards? This is a fairly common question because the RL framework is often criticized for requiring a handcrafted reward function, which is often difficult to provide for many problems of interest. To understand our affirmative answer to this question, it is necessary to do a little deconstruction of the RL framework.

### B. Internal and External Environments

According to the "standard" RL framework [19], the agent-environment interaction is envisioned as the interaction between a controller (the agent) and a controlled system (the environment), with a specialized reward signal coming

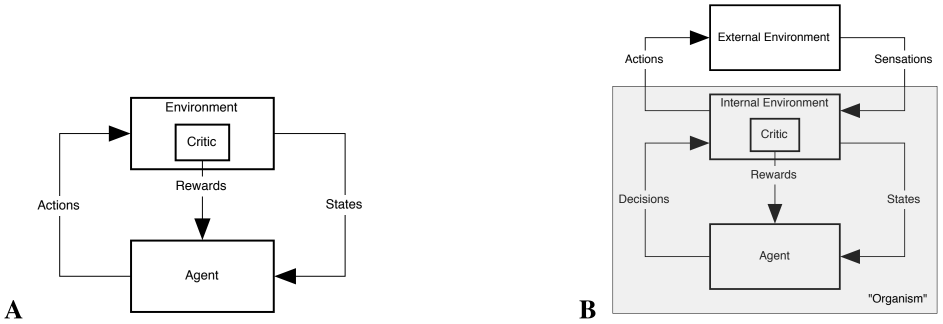

Fig. 1. *Agent-Environment Interaction in Reinforcement Learning.* **A***: Reward is supplied to the agent from a "critic" in its environment.* **B***: An elaboration of Panel A in which the environment is factored into an internal and an external environment, with reward coming form the former. The shaded box corresponds to what we would think of as the "organism."*

from the environment to the agent that provides at each moment of time a scalar reward value. The component of the environment providing the reward signal is usually called the "critic" (Fig. 1A). The agent learns how to increase the total amount of reward it receives over time from the critic. With appropriate mathematical assumptions, the problem faced by the learning agent is that of approximating an optimal policy for a Markov decision process (MDP).

Sutton and Barto [19] point out that the scheme in Fig. 1A is very abstract, and that one should not identify this agent with an entire animal or robot. An animal's reward signals are determined by processes within its brain that monitor not only external events through exteroceptive systems but also the animal's internal state, which includes information pertaining to critical system variables (e.g., blood-sugar level, core temperature, etc.) as well as memories and accumulated knowledge. The critic is in an animal's head. Fig. 1B makes this more explicit by factoring the environment of Fig. 1A into an *external environment* and an *internal environment*, the latter of which contains the critic responsible for generating primary reward. Notice that this scheme still includes cases in which reward can be thought of as an external stimulus (e.g., a pat on the head or a word of praise). These are stimuli transduced by the internal environment so as to generate appropriate reward signals.

Because Fig. 1B is a refinement of Fig. 1A (that is, it is the result of adding structure rather than changing it), the standard RL framework already encompasses intrinsic reward. In fact, according to this model, *all* reward is intrinsic.

### *C. Problem-Independent Rewards and Competence*

If we accept Fig. 1B and the view that all rewards are intrinsically generated, then what do we mean by intrinsically-motivated RL? Our view is that the key distinction is between problem-specific and problem-independent reward functions. We define extrinsic reward to be the result of a problem-specific reward function. By designing such a function, we can "motivate" an RL system to learn how to solve a particular problem, such as how to play backgammon, how to dispatch elevators, etc. Intrinsic reward, on the other hand, is reward generated by a reward function—an intrinsic reward function—designed to facilitate learning a wide class of problems instead of a particular one. Intrinsic rewards can motivate efficient exploration, efficient model building, efficient hypothesis formation and testing, and other behaviors that are generally useful for acquiring knowledge needed to solve a range of specific problems. Indeed, the very act of learning itself might be intrinsically rewarding, as suggested by a number of researchers (e.g., [10]).

Whenever intrinsic and extrinsic reward functions are simultaneously in force it is important to consider how they interact. Intrinsic rewards can disrupt learning to extrinsic rewards because they effectively re-define the problem that the learning agent is trying to solve. An agent may pursue its intrinsic goals at the expense of achieving its extrinsically defined goals One way to address this problem is to make sure that intrinsic rewards are transient so that the problem eventually reverts to the extrinsically specified one. A related approach is to separate learning into a *developmental phase* during which only intrinsic rewards are generated, and a *mature phase* during which only extrinsic rewards are generated. While a strict separation is neither realistic nor necessary, it provides a simple framework in which to study intrinsic reward systems.

We are guided by White's classic paper [21] where it is argued that intrinsically-motivated behavior is essential for an organism to gain the *competence* necessary for autonomy. A system that is competent in this sense has a broad set of reusable skills for controlling its environment. The activity through which these broad skills are learned is motivated by an intrinsic reward system that favors the development of broad competence rather than being directed to more specific externally-directed goals. These skills act as the "building blocks" out of which an agent can form solutions to specific problems that arise over its lifetime. Instead of facing each new challenge by trying to create a solution out of low-level primitives, it can focus on combining and adjusting higher-level skills, greatly increasing the efficiency of learning to solve new problems.

## III. SKILLS

What do we mean by a skill? Recent RL research provides a concrete answer to this question, together with a set of algorithms capable of improving skills with experience. To combat the complexity of learning in difficult domains, RL researchers have developed ways of exploiting "temporal abstraction," where decisions are not required at each step, but rather where each decision invokes the execution of a temporally-extended activity that follows its own closed-loop policy until termination. Substantial theory exists on how to plan and learn when temporally-extended skills are added to the set of actions available to an agent. Since a skill can invoke other skills as components, hierarchical control architectures and learning algorithms naturally emerge from this conception of a skill. Specifically, our approach builds on the theory of *options* [20], and below we use the terms skill and option interchangeably.

### A. *Options*

A brief account of the option framework follows, which starts with a finite MDP. At each stage in a sequence of stages, an agent observes a system's state, $s$, contained in a finite set, $\mathcal{S}$, and executes an action, $a$, selected from a finite, non-empty set, $\mathcal{A}_s$, of admissible actions. The agent receives an immediate reward having expected value $R(s,a)$, and the state at the next stage is $s'$ with probability $P(s'|s,a)$. The expected immediate rewards, $R(s,a)$, and the state transition probabilities, $P(s'|s,a)$, $s, s' \in \mathcal{S}$, $a \in \mathcal{A}_s$, together comprise the *one-step model* of action $a$. A (stationary, stochastic) policy $\pi : \mathcal{S} \times \cup_{s\in\mathcal{S}} \mathcal{A}_s \rightarrow [0,1]$, with $\pi(s,a) = 0$ for $a \notin \mathcal{A}_s$, specifies that the agent executes action $a \in \mathcal{A}_s$ with probability $\pi(s,a)$ whenever it observes state $s$. The objective is to find a policy that maximizes the expected return from each state, where return is a function of future rewards and can be defined in a number of different ways [3].

Starting from a finite MDP, which we call the *core* MDP, the simplest kind of option $o$ consists of a policy $\pi^o : \mathcal{S}\times \cup_{s\in\mathcal{S}} A_s \rightarrow [0,1]$, a termination condition $\beta^o : \mathcal{S} \rightarrow [0,1]$, and an input set $\mathcal{I}^o \subseteq \mathcal{S}$. The option $o = \langle \mathcal{I}^o, \pi^o, \beta^o \rangle$ is available in state $s$ if and only if $s \in \mathcal{I}^o$. If the option is executed, then actions are selected according to $\pi^o$ until the option terminates stochastically according to $\beta^o$. For example, if the current state is $s$, the next action is $a$ with probability $\pi^o(s,a)$, the environment makes a transition to state $s'$, where the option either terminates with probability $\beta^o(s')$ or else continues, determining the next action $a'$ with probability $\pi^o(s',a')$, and so on. When the option terminates, the agent can select another option from the set of those available at the termination state. Note that any action of the core MDP, a *primitive action* $a \in \cup_{s\in\mathcal{S}}\mathcal{A}_s$, is also an option, called a *one-step option*, with $\mathcal{I} = \{s : a \in \mathcal{A}_s\}$ and $\beta(s) = 1$ for all $s \in \mathcal{S}$.

A policy $\mu$ over options selects option $o$ in state $s$ with probability $\mu(s,o)$; $o$'s policy in turn selects other options until $o$ terminates. The policy of each of these selected options selects other options, and so on, until one-step options are selected that correspond to actions of the core MDP. Adding any set of options to a core finite MDP yields a well-defined discrete-time semi-Markov decision process whose actions are the options and whose rewards are the returns delivered over the course of an option's execution.

One can define value functions corresponding to options in a manner analogous to how they are defined for simple MDPs. For example, the option-value function corresponding to $\mu$ is defined as follows:

$$Q^\mu(s,o) = E\{r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{\tau-1} r_{t+\tau} + \cdots | \mathcal{E}(o\mu, s, t)\},$$

where $\mathcal{E}(o\mu,s,t)$ is the event of $o$ being initiated at time $t$ in $s$ and being followed until it terminates after $\tau$ time steps, at which point control continues according to $\mu$.

A multi-time model of an option, which we call an *option model*, generalizes the one-step model of a primitive action. For any option $o$, let $\mathcal{E}(o,s,t)$ denote the event of $o$ being initiated in state $s$ at time $t$. Then the reward part of the option model of $o$ for any $s \in S$ is:

$$R(s,o) = E\{r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{\tau-1} r_{t+\tau} | \mathcal{E}(o,s,t)\},$$

where $t+\tau$ is the random time at which $o$ terminates. The state-prediction part of the model of $o$ for $s$ is:

$$P(s'|s,o) = \sum_{\tau=1}^{\infty} p(s',\tau)\gamma^\tau,$$

for all $s' \in \mathcal{S}$, where $p(s',\tau)$ is the probability that $o$ terminates in $s'$ after $\tau$ steps when initiated in $s$. Though not itself a probability, $P(s'|s,o)$ is a combination of the probability that $s'$ is the state in which $o$ terminates together with a measure of how delayed that outcome is in terms of $\gamma$.

The quantities $R(s,o)$ and $P(s'|s,o)$ respectively generalize the reward and transition probabilities, $R(s,a)$ and $P(s'|s,a)$, of the usual MDP in such a way that it is possible to write a generalized form of the Bellman optimality equation and extend RL methods to options. In the work reported here we use *intra-option learning methods*, which allow the policies of many options to be updated simultaneously during an agent's interaction with the environment. If an option *could have* produced a primitive action in a given state, its policy can be updated on the basis of the observed consequences even though it was not directing the agent's behavior at the time. Related methods have been developed for learning option models of many options simultaneously by exploiting Bellman-like equations relating the components of option models for successive states [20].

### B. *Intrinsic Rewards and Options*

The connection between intrinsic motivation and options, first presented in refs. [1] and [17], is the idea of creating an option upon the occurrence of an intrinsically-rewarding event, where what constitutes an intrinsically-rewarding event can be defined in numerous ways to be described shortly. Many researchers have recognized the desirability of automatically creating options (e.g., refs. [6], [8], [13]). In these approaches, some means is devised for identifying states that may usefully serve as "subgoals" for a given task. An option is created whose policy, when it is fully learned, will control the environment to a subgoal state in an efficient manner, usually in minimum time, from any state in the option's input set, which may itself be learned. The option's termination condition is set to be the achievement of a subgoal state, and its policy is learned via a "pseudo reward function" [5] which rewards the achievement of the subgoal and provides a small penalty to all other transitions. This is a *pseudo* reward function because it is distinct from the reward function that defines the agents overall task, and it does not directly influence the behavior of the agent. It is used only to support the learning of the option's policy,

Intrinsic motivation enters into this picture in two ways. First, psychological studies of intrinsic motivation provide numerous guidelines as to what should constitute problem-independent intrinsically-rewarding events. Berlyne [2] probably had the most to say on these issues, suggesting that the factors underlying intrinsic motivational effects involve novelty, surprise, incongruity, and complexity. He also hypothesized that moderate levels of novelty have the highest hedonic value because the rewarding effect of novelty is overtaken by an aversive effect as novelty increases. This is consistent with many other views holding that situations intermediate between complete familiarity (boredom) and complete unfamiliarity (confusion) have the most hedonic value. Another hypothesis about what we find satisfying in exploration and manipulation is that we enjoy "being a cause" [7], which is a major component of Piaget's theory of child development [14]. In this paper, we use only the degree of surprise of salient stimuli as intrinsic reward, but this is merely a starting point.

The second way intrinsic motivation adds to earlier option-creation ideas is that, unlike pseudo rewards, intrinsic rewards influences agent behavior. The agent should change its behavior in such as way that it focuses exploration in order to quickly refine its skill in bringing about the intrinsically-rewarding event. This is what motivation means: the agent has to "want" to bring about the event in question, and this has to be manifested in its behavior. Pseudo reward functions do not do this. A corollary to this is that intrinsic reward should diminish with continued repetition of the activity that generates it, i.e., the agent should eventually get bored and move on to create and learn another option.

## IV. EXAMPLE

We briefly describe an example implementation of some of these ideas in a simple artificial "playroom" domain. See refs. [1], [17] for details. In the playroom (a 5x5 grid), are a number of objects: a light switch, a ball, a bell, two movable blocks that are also buttons for turning music on and off, as well as a toy monkey that can make sounds. The agent has an eye, a hand, and a visual marker. At any time step, the agent has a collection of actions available to it, such as: move eye to hand, move eye to marker, move eye one step north, south, east or west, etc. In addition, if both the eye and and hand are on some object, then natural operations suggested by the object become available, e.g., if both the hand and the eye are on the light switch then the action of pushing the light switch becomes available. The objects in the playroom all have potentially interesting characteristics. The bell rings once and moves to a random adjacent square if the ball is kicked into it. The light switch controls the lighting in the room. The color of any of the blocks in the room is only visible if the light is on, otherwise they appear similarly gray. The blue block if pressed turns music on, while the red block if pressed turns music off. The toy monkey makes frightened sounds if simultaneously the room is dark and the music is on and the bell is rung.

These objects were designed to have varying degrees of difficulty to engage. For example, to get the monkey to cry out requires the agent to do the following sequence of actions: 1) get its eye to the light switch, 2) move hand to eye, 3) push the light switch to turn the light on, 4) find the blue block with its eye, 5) move the hand to the eye, 6) press the blue block to turn music on, 7) find the light switch with its eye, 8) move hand to eye, 9) press light switch to turn light off, 10) find the bell with its eye, 11) move the marker to the eye, 12) find the ball with its eye, 13) move its hand to the ball, and 14) kick the ball to make the bell ring. If the agent has already learned how to turn the light on and off, how to turn music on, and how to make the bell ring, then those learned skills would be of obvious use in simplifying this process of engaging the toy monkey.

For this simple example, the agent has a built-in notion of salience of stimuli. In particular, changes in light and sound intensity are considered salient by the playroom agent. The agent behaves by choosing actions according to a value function [19]. The agent starts by exploring its environment randomly. Each first encounter with a salient event initiates the learning of an option and an option model with that salient event as its goal. For example, the first time the agent happens to turn the light on, it initiates the data-structures necessary for learning and storing the light-on option, including the initiation set, the policy, the termination probabilities, as well as for storing the light-on option model. As the agent moves around the world, all the options and their models are simultaneously updated using intra-option learning algorithms.

We experimented with two methods for providing intrinsic reward to the agent: rewarding errors in prediction of salient events and rewarding certain changes in option models. The first method was suggested by the novelty response of dopamine neurons [9] and proposed in refs. [1] and [17]. The intrinsic reward for each salient event is proportional to the error in the agent's prediction of that salient event according to the current option model corresponding to that event. The intrinsic reward is used to

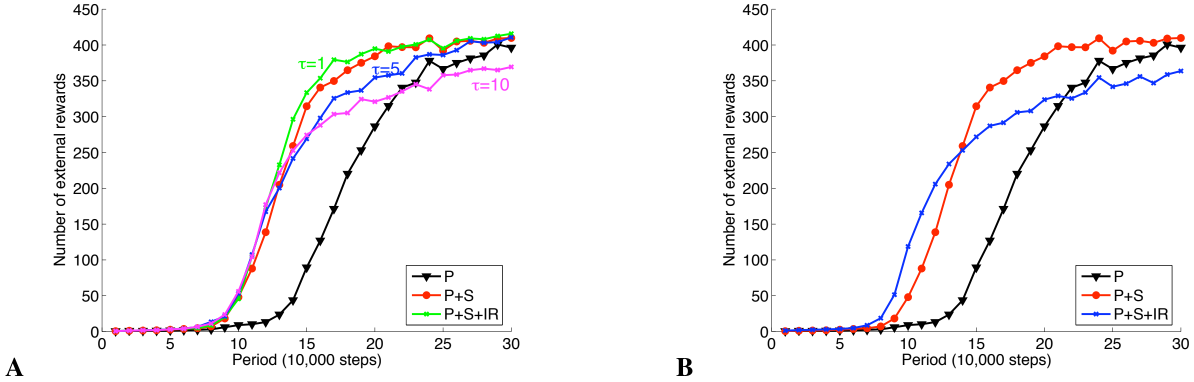

Fig. 2. Number of extrinsic rewards received over time. **A.** Intrinsic reward is a function of prediction error. P: primitive actions only; P+S: primitive actions and skills; P+S+IR: primitive actions, skills, and intrinsic reward. Parameter $\tau$ determines relative weight of intrinsic and extrinsic reward: larger $\tau$ means higher weight for intrinsic reward. **B.** Intrinsic reward is a function of changes in option models; $\tau = 5$.

update the value function the agent is using to determine its behavior in the playroom. As a result, when the agent encounters an unpredicted salient event a few times, its updated value function drives it to repeatedly attempt to achieve that salient event. There are two interesting side effects of this: 1) as the agent repeatedly tries to achieve the salient event, learning improves both its policy for doing so and its option model that predicts the salient event, and 2) as its option policy and option model improve, the intrinsic reward diminishes and the agent gets "bored" with the associated salient event and moves on. Occasionally, the agent encounters the salient event in a state that it has not visited before, and it generates intrinsic reward again (it is "surprised").

We looked at a playroom task in which extrinsic reward was available only if the agent succeeded in making the monkey cry out. We compared the performance of three agents. One agent learned using only the primitive actions, another learned a skill hierarchy using pseudo rewards but not intrinsic reward, and the third agent learned a skill hierarchy and also generated intrinsic reward. Our results (which space does not permit us to present) clearly show the great advantage of the skill hierarchy, but the addition of intrinsic reward did not improve performance. In fact, weighting intrinsic reward heavily compared to extrinsic reward actually hampered learning. Further analysis revealed two shortcomings of the algorithm. First, the intrinsic reward is too persistent, meaning that the agent attaches a high value to certain actions long after the intrinsic reward for those actions has disappeared. And second, the intrinsic reward does not propagate well, tending to remain restricted to the immediate vicinity of the salient event that gave rise to it.

In the second method, the agent receives intrinsic reward for certain types of changes in its option models. Let $P_t(s'|s,o)$ be the model of any option $o$ at step $t$; let $\mathcal{G}^o$ be the set of states at which the goal of $o$ is achieved; and let $\mathcal{I}^o$ be the initiation set of $o$. Then, $r^i_t$, the intrinsic reward at step $t$, is the following:

$$r^i_t = \sum_{\forall o} \sum_{\substack{\forall s' \in \mathcal{G}^o \\ \forall s \in \mathcal{I}^o}} P_t(s'|s,o) - P_{t-1}(s'|s,o) \qquad (1)$$

where the outer sum is taken over all the agent's options.

The intuition behind this is that the policy for an option should take the agent to the option's goal region (for example, to a state where the light is on) as quickly as possible. In other words, $P(s'|s,o)$ should be as large as possible for $s'$ in the goal region. Recall that $P(s'|s,o)$ is a sum of discounted probabilities. The largest value it can take is 1, which models an option that reliably takes the agent to state $s'$ in one step from state $s$; the more reliable the option is in terminating at $s'$ and the fewer the number of transitions it takes to achieve this, the closer its value will be to 1. The intrinsic reward of Eq. 1 rewards changes in $P(s'|s,o)$ toward this ideal. As a result, actions that allow the agent to improve its option policies will be rewarded. Not only will the agent show a preference for such actions, it will seek them out, exploring those regions of the environment where it expects to learn how to improve its subgoal-seeking behavior.

We present preliminary results in a simplified version of the playroom domain in which there is extrinsic reward for making the monkey cry out and there is only one salient event: turn music on. The agent learns the single skill for turning on the music. Fig. 2A shows the number of extrinsic rewards obtained over an extended time period, comparing the performance of three agents. One agent (P) learns using only the primitive actions, another agent (P+S) learns a skill hierarchy but does not generate intrinsic reward, and the third agent learns a skill hierarchy and also generates intrinsic reward (P+S+IR). Multiple curves are shown for P+S+IR, each for a different value of the algorithm's key parameter, $\tau$, which determines the relative weight of the intrinsic reward compared to the extrinsic reward. The larger the value of $\tau$, the larger the relative weight placed upon intrinsic reward. The figure, which shows mean values over 50 repetitions of the experiment, clearly shows the advantage of the skill hierarchy over the use of just primitive actions. The intrinsic reward, however, is not effective, improving performance only incrementally. For the values of $\tau$ with which we experimented, performance decreased with increasing $\tau$.

Fig. 2B shows results using intrinsic reward defined by Eq. 1. The intrinsic reward function had a positive influence in the early stages of learning, but later provided a distraction for the agent, as the agent continued to be rewarded for both changes in the option model and for making the monkey cry out. Repeated experiments with a range of values for $\tau$ gave qualitatively similar results; the figure shows performance with $\tau = 5$.

## V. CONCLUSION

These experimental results—only very briefly presented due to space limitations—support previous research in showing that the construction of temporally-extended skills formulated as options can confer clear advantages over learning solely with primitive actions. On the other hand, these results also show that defining an effective form of intrinsic reward is not as straightforward as we had at first thought. Intrinsic reward can reduce the speed of learning by making the agent persist in behavior directed toward a salient event long after that behavior has been well learned. This kind of "obsessive-compulsive" behavior hinders the attainment of extrinsic goals (though we are not ready to propose this as a theory of OCD!). In addition, intrinsic reward does not propagate well, tending to remain restricted to the immediate vicinity of the salient event that gave rise to it. There are many possibilities for addressing these problems.

A wider view, however, suggests that it is not adequate to assess the impact of intrinsically-motivated learning in terms of its effect on learning a specific extrinsically motivated task. The view we have put forward for the benefits of intrinsically-motivated behavior is that it serves to build a repertoire of skills that can be useful across *many* future extrinsically-motivated tasks. One might expect, and our personal experience indeed tends to bear this out, that intrinsically-motivated behavior will take a toll in terms of immediate solutions to specific extrinsic tasks. This is the exploration-exploitation dilemma at a somewhat larger scale than we are used to thinking about in RL.

Finally, we emphasize that our definition of intrinsic reward in terms of a pre-defined set of salient events is only one of the simplest possibilities. Many additional definitions are suggested by the psychological literature, previous computational research on intrinsic motivation, as well as research on the neuroscience of brain reward systems. We expect that there is a wide assortment of situations, defined in terms of both an agent's external and internal environments, that will form the basis for richer forms of intrinsic reward.

## VI. ACKNOWLEDGMENTS

The authors thank Satinder Singh and Nuttapong Chentanez for their previous work on this topic. This research was funded by NSF grant CCF 0432143 and by a grant from DARPA's IPTO program.

## REFERENCES

[1] A. G. Barto, S. Singh, , and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the International Conference on Developmental Learning (ICDL)*, 2004.

[2] D. E. Berlyne. *Conflict, Arousal. and Curiosity*. McGraw-Hill, N.Y., 1960.

[3] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[4] P. Dayan. Motivated reinforcement learning. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Conference*, pages 11–18, Cambridge MA, 2001. MIT Press.

[5] T. G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[6] B. Digney. Learning hierarchical control structure from multiple tasks and changing environments. In *From Animals to Animats 5: The Fifth Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1998. MIT Press.

[7] K. Groos. *The Play of Man*. D. Appleton, N.Y., 1901.

[8] B. Hengst. Discovering hierarchy in reinforcement learning with HEXQ. In *Maching Learning: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 243–250, San Francisco, CA, 2002. Morgan Kaufmann.

[9] J. C. Horvitz. Mesolimbocortical and nigrostriatal dopamine responses to salient non-reward events. *Neuroscience*, 96:651–656, 2000.

[10] F. Kaplan and P.-Y. Oudeyer. Maximizing learning progress: An internal reward system for development. In F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi, editors, *Embodied Artificial Intelligence*, pages 259–270. Springer-Verlag, 2004.

[11] D. B. Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics*. PhD thesis, Stanford University, 1976.

[12] S. M. McClure, N. D. Daw, and P. R. Montague. A computational substrate for incentive salience. *Trends in Neurosciences*, 26:423–428, 2003.

[13] A. McGovern and A. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 361–368, San Francisco, CA, 2001. Morgan Kaufmann.

[14] J. Piaget. *The Origins of Intelligence in Children*. Norton, N.Y., 1952.

[15] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 222–227, Cambridge, MA, 1991. MIT Press.

[16] W. Schultz, P. Dayan, and P. R. Montague. A neural substrate of prediction and reward. *Science*, 275:1593–1598, March 1997.

[17] S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, Cambridge MA, 2005. MIT Press. To appear.

[18] R. S. Sutton. Learning to predict by the method of temporal differences. *Machine Learning*, 3:9–44, 1988.

[19] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[20] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[21] R. W. White. Motivation reconsidered: The concept of competence. *Psychological Review*, 66:297–333, 1959.