Internal Rewards Mitigate Agent Boundedness

Jonathan Sorg JDSORG@UMICH.EDU
Satinder Singh BAVEJA@UMICH.EDU
Computer Science & Engineering, University of Michigan, 2260 Hayward Street, Ann Arbor, MI 48109

Richard Lewis RICKL@UMICH.EDU
Department of Psychology, University of Michigan, 530 Church Street, Ann Arbor, MI 48109

Abstract

Reinforcement learning (RL) research typically develops algorithms for helping an RL agent best achieve its goals—however they came to be defined—while ignoring the relationship of those goals to the goals of the agent designer. We extend agent design to include the meta-optimization problem of selecting internal agent goals (rewards) which optimize the designer's goals. Our claim is that well-designed internal rewards can help improve the performance of RL agents which are computationally bounded in some way (as practical agents are). We present a formal framework for understanding both bounded agents and the meta-optimization problem, and we empirically demonstrate several instances of common agent bounds being mitigated by general internal reward functions.

1. Introduction

Consider the problem of an agent designer building an artificial agent to act autonomously in some sequential decision making environment. The designer's goals and purposes implicitly define a preference over possible behaviors of the agent. In most complex and partially known environments, the optimal behavior of the agent will not be known to the designer and hence cannot be hardwired into the agent. Instead, a common and often more-robust approach is to build into the agent its own *internal* goals (along with other knowledge that the designer has).

Should the artificial agent's goals be the same as the agent designer's goals? This question is seldom asked. Rather, most research on autonomous agents focuses on algorithms and architectures for achieving agent goals—however they came to be defined. Yet, there is always an agent designer with his or her own goals, and they have to be translated somehow into agent goals. Confounding the agent's and designer's goals, i.e., assuming that they are the same, is of course a powerful and simple constraint on agent design, and so there is a cost to violating this assumption in terms of increased complexity of the design process. It is therefore important that we understand when and why violating this assumption—what we call *breaking the confound*—may improve agent performance with respect to the designer's goals. In this paper, we take first steps toward formalizing and empirically supporting a claim about the conditions under which breaking the confound is beneficial: optimizing internal rewards can lead to improved performance in RL agents with computational bounds. In such cases of improved performance, we say that *internal rewards mitigate agent boundedness.* We can state this claim more strongly as a directive for agent design: because agents are often limited in some way, they should often be given goals (rewards) that are *distinct* from the designer's goals, and the nature of these distinct goals depends directly on the nature of the bounds.

In the main body of this paper we provide several empirical illustrations supporting the generality of this claim by showing how internal rewards mitigate specific aspects of boundedness that arise naturally in current practice in the design of both model-free and model-based learning agents. These bounds include limits on planning depth, limits on state or model representation, and limits arising from choices of function approximation and learning algorithms. To help sharpen our claims and provide a general framework in which to understand our empirical results, we turn next to defining unbounded and bounded agents and performance gaps between them.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

2. On Boundedness & Performance

We first briefly define the environment dynamics and the unconventional separation of agent designer's goals and the RL agent's goals, then identify performance differences between classes of agents.

Dynamics. At each time step, an agent G receives an observation $o \in \mathcal{O}$ from its environment M, takes an action $a \in \mathcal{A}$, and repeats this process until a time horizon. The state at time step k, denoted h_k, is the history of interaction $o_1 a_1 \cdots o_{k-1} a_{k-1} o_k$ to time step k. The transition dynamics define a probability distribution over possible next states h_{k+1} as a function of h_k and a_k. Although representations of state more compact than full histories are used in practice, for this discussion it is convenient to use history as state for complete generality.

Agent Designer's Goals. We represent the agent designer's goals using an *objective reward function* $R_\mathcal{E}$ that maps states to scalar reward values, and a cumulative measure, $F_\mathcal{E}$, that defines how the reward along a trajectory of states is accumulated. Common choices for $F_\mathcal{E}$ include summed reward over a finite (fixed or random) horizon, and the average or discounted reward over an infinite horizon. Together, $R_\mathcal{E}$ and $F_\mathcal{E}$ define an *objective return function* $F_{R_\mathcal{E}}$ which in the average reward case, for example, defines the return for history h_∞, as $F_{R\mathcal{E}}(h_\infty) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} R_\mathcal{E}(h^i_\infty)$, where h^i_k is the i^{th} state in a history of length k. The designer's goal then is to design an agent whose behavior maximizes the expected objective return.

RL Agent's Goals. Our main departure from the conventional framing of the RL problem is allowing the agent's goals to be defined independently of the designer's goals. Specifically, the designer sets an *internal* reward R_I and cumulative measure F_I for the agent which together define an internal return function F_{R_I}. We will refer to the conventional setting of $R_I = R_\mathcal{E}$ and $F_{R_I} = F_{R_\mathcal{E}}$ as *confounded* reward and return functions respectively. The RL agent's goal is to behave so as to maximize its expected internal return, regardless of how it came to be defined.

The design space of agents. Extensionally, an agent G is a mapping from histories to (possibly distributions over) actions. Intensionally, agents are defined via some parameters such as learning and exploration rates for Q-learning agents, planning depth for look-ahead-based planning agents, etc. Let *all* the conventional parameters of agent design be denoted θ; thus, in conventional RL $G(\theta)$ fully specifies an agent. The *unbounded agent parameter space*, denoted by $\bar{\Theta}$, corresponds to the most general space of agents—extensionally, the space of all possible mappings from histories to actions. In most practical cases, because of computational resource constraints, the designer's search for good agents will be limited to some much smaller *bounded* set of agent parameters denoted Θ, such that $\Theta \subset \bar{\Theta}$.

In our new framing of the problem, the internal return function F_{R_I} is another important parameter of agent design; thus $G(F_{R_I}, \theta)$ fully specifies an agent in this paper. The internal reward function is chosen from a set of possible internal return functions $\mathcal{F_R}$, hereafter assumed to contain $F_{R_\mathcal{E}}$. While $\mathcal{F_R}$ may also be bounded or unbounded, this distinction is not needed for our claims and henceforth we let it be whatever (bounded) set is available to the designer. Any specific agent $G(F_R, \theta)$ in the environment M will produce a distribution over histories $h \sim \langle G(F_R, \theta), M \rangle$.

To sharpen our notions of performance differences between bounded and unbounded agents, it is useful to consider the six types of agents defined in Table 1. Each agent, labeled A–F, is defined as a solution to a meta-optimization problem constrained by choices of reward function space $\mathcal{F_R}$ and agent parameter space Θ. The quantity optimized is always the expected objective return $F_{R_\mathcal{E}}$, though only agents A, B and C are constrained to use $F_{R_\mathcal{E}}$ internally.

The *conventional agent* A $= G(F_{R_\mathcal{E}}, \theta)$ is the usual agent in an RL setting where θ are the agent parameters the designer favors, usually based on prior experience or partial search over some very limited Θ. It is a designer's attempt at building a *θ-optimal agent* B. Both A and B agents are given the (confounded) objective return. Finding the true θ-optimal agent for rich Θ is usually intractable. The *θ-unbounded* agent C is the best agent in principle that uses the objective return. An *F-optimal agent* D is the agent that breaks the confound by optimizing internal reward but uses whatever conventional parameters θ the designer prefers. A *boundedly optimal agent* E is the best an agent designer could hope to achieve; it is a solution to the joint optimization over the space $(\mathcal{F_R}, \Theta)$. The parameters θ for agents C and F are the best from the unbounded set $\bar{\Theta}$; the parameters for agents B and E are the best from the bounded set Θ. There is no better agent than the *unbounded* agent F.

The performance gaps among agents. It should be clear from the definitions in Table 1 that the following partial ordering holds among the performance of agents: A$\preceq$B$\preceq$C$\preceq$F, A$\preceq$D$\preceq$E$\preceq$F, and B$\preceq$E. The performance difference or *gap* of primary theoretical interest in this paper is the difference between the θ-optimal agent B and the θ-unbounded agent C. With-

Table 1. Six types of agents constructed as solutions to meta-optimization problems defined by constraints on the internal reward function space (rows) and the agent's conventional parameter space (columns).

	Fixed θ $G(\cdot,\theta)$	*Bounded* $G(\cdot,\theta^* \in \Theta)$	*Unbounded* $G(\cdot,\bar{\theta}^* \in \bar{\Theta})$
Confounded reward $G(F_{R_\mathcal{E}},\cdot)$	(A) Conventional agent $G(F_{R_\mathcal{E}},\theta)$	(B) θ-optimal agent $G(F_{R_\mathcal{E}},\theta^*_\mathcal{E})^b$	(C) θ-unbounded agent $G(F_{R_\mathcal{E}},\bar{\theta}^*_\mathcal{E})^c$
Unconfounded reward $G(F^*_{R_I} \in \mathcal{F}_\mathcal{R},\cdot)$	(D) F-optimal agent $G(F^*_{R_I},\theta)^d$	(E) Boundedly optimal agent $G(F^*_{R_I},\theta^*_I)^e$	(F) Unbounded agent $G(F^*_{R_I},\bar{\theta}^*_I)^f$

[b] $\theta^*_\mathcal{E} = \arg\max_{\theta\in\Theta} \mathbb{E}[F_{R_\mathcal{E}}(h)|h \sim \langle G(F_{R_\mathcal{E}},\theta), M\rangle]$
[c] $\bar{\theta}^*_\mathcal{E} = \arg\max_{\bar{\theta}\in\bar{\Theta}} \mathbb{E}[F_{R_\mathcal{E}}(h)|h \sim \langle G(F_{R_\mathcal{E}},\bar{\theta}), M\rangle]$
[d] $F^*_{R_I} = \arg\max_{F_R\in\mathcal{F}_\mathcal{R}} \mathbb{E}[F_{R_\mathcal{E}}(h)|h \sim \langle G(F_R,\theta), M\rangle]$
[e] $(F^*_{R_I},\theta^*_I) = \arg\max_{F_R\in\mathcal{F}_\mathcal{R};\theta\in\Theta} \mathbb{E}[F_{R_\mathcal{E}}(h)|h \sim \langle G(F_R,\theta), M\rangle]$
[f] $(F^*_{R_I},\bar{\theta}^*_I) = \arg\max_{F_R\in\mathcal{F}_\mathcal{R};\bar{\theta}\in\bar{\Theta}} \mathbb{E}[F_{R_\mathcal{E}}(h)|h \sim \langle G(F_R,\bar{\theta}), M\rangle]$

out breaking the confound, B is the best agent the designer could build within bounds and C is the best agent the designer could have built without bounds.

The title of this paper, *internal rewards mitigate agent boundedness*, refers to our conjecture that for many environments and available parameters Θ, the performance of the boundedly-optimal agent E will be *strictly greater* than that of the θ-optimal agent B; that is, the performance of E mitigates the gap between bounded agent B and unbounded agent C: B$\prec$E$\preceq$C. It is intractable, however, to find θ^*. Thus, in our empirical work, we will approximate the demonstration of mitigation of the performance gap of interest by replacing the θ-optimal agent B with the conventional agent A, and the boundedly optimal agent E with our best approximation of the F-optimal agent D. Better choices of θ in the conventional agent yield tighter approximations of the gap of interest.

We conjecture that there is no performance difference between E and F, because unbounded agents will be able to optimally maximize confounded return and thus there will be no need to break the confound. The implications of this conjecture are important: as the set of available agents expands (Θ approaches $\bar{\Theta}$), the opportunity for internal rewards to help diminishes.

3. Empirical Demonstrations

We describe here five experiments designed to illustrate the mitigation of agent bounds by internal reward. The first four experiments involve model-based learning agents in foraging domains with bounds on planning depth, state representation, or model representation. The fifth experiment, a version of the *acrobot* problem (Sutton & Barto, 1998), extends our empirical work to domains not constructed by us, and to learning agents with function approximation.

Each experiment has the same structure: (1) We investigate the performance of an agent with some specific bound in an environment which reveals the performance limitations that arise from that bound. Such agents are instances of the conventional agents A in Table 1. (2) We formulate a space of internal rewards as a scalar mapping from (mostly M-independent) features that are motivated by their potential to overcome the bounds represented by θ. (3) Through search of this space by directly simulating agents in their environments[1], we approximate $F^*_{R_I}$ and compare the performances of the approximate F-optimal agent $G(\hat{F}^*_{R_I},\theta)$, the agent with confounded reward, and the unbounded agent (if available). In some cases, we are able to do (1)–(3) while also varying the agent bound parametrically.

3.1. Experiment 1: Mitigating bounds on planning depth

The objective of this experiment is to show that internal rewards based on recency features (how recently each action has been taken in each state) can mitigate limitations of bounded planning depth by encouraging systematic exploration. We will also provide an empirical demonstration that an unbounded agent may not benefit from breaking the confound.

The environment and designer's goals. Consider the 3-Corridor foraging environment illustrated in Figure 1(a). The environment consists of a 3 × 3 grid world with 3 dead-end corridors (rows) separated by impassable walls. The (bird) agent has four available

[1] Our optimization procedure adaptively samples reward vectors in the unit sphere, as it can be shown that for the (linear) form of the reward functions and for the agents presented here, searching this subset is equivalent to searching the entire space. We further approximate infinite-horizon average returns with returns over long finite horizons.

actions which deterministically move the agent in each of the cardinal directions. If the intended direction is blocked by a wall or the boundary, the action results in no movement. There is a (worm) food source randomly located in one of the three right-most locations at the end of each corridor. The agent has an `eat` action, which consumes the worm when the agent is at the worm's location. After the agent consumes the worm, the agent becomes *satiated* for 1 time step, and the worm disappears. Immediately, a new worm appears randomly in one of the other two potential worm locations. At all other time steps, the agent is *hungry*. The agent observes the entire state: the agent's location, whether it is hungry, and the worm's location.

The designer's goal is to maximize worms eaten. Thus, the objective reward function $R_{\mathcal{E}}$ provides a reward of 1.0 when the agent eats a worm (i.e., is satiated) in the current observation, and a reward of 0 otherwise. We use the infinite-horizon average return function.

The agent and its bounds. Let G_d (short for $G(R_I, d)$) denote a *depth-d planning model-based learning agent*—an agent that acts greedily with respect to the d-step action-value function $Q_d(s,a) = \sum_{s' \in S} \hat{T}(s'|s,a)[R_I(s,a,s') + \gamma \max_{a'} Q_{d-1}(s',a')]$ where $Q_0(s,a) \overset{\text{def}}{=} 0$. (The agents use $\gamma = 0.99$ unless noted otherwise.) If the values of multiple actions are equivalent, the agent selects randomly among them. The transition-dynamics estimates, $\hat{T}$, come from an estimated MDP transition model (updated after every action) based on the empirical transition probabilities between assumed-Markov observations. Specifically, let $n_{s,a}$ be the number of times that action a was taken in state s. Let $n_{s,a,s'}$ be the number of times that s' was reached after taking action a in state s. The agent models the probability of reaching s' after taking a in state s as $\hat{T}(s'|s,a) = \frac{n_{s,a,s'}}{n_{s,a}}$.[2]

Thus, agent G_d is a simple example of a computation-bounded agent in which the depth d is a parameter controlling the degree of boundedness. More specifically, agent G_0 is a random agent, because its Q-function is a constant 0; agent G_1 acts greedily with respect to its reward, and agent G_∞ is an unbounded-depth planning agent, computing the optimal value function with respect to its current model and internal reward function. In experiment 1's environment, the largest (over all states) look-ahead needed to obtain objective reward is 8. Thus we explore agents with planning depths between 0 and 9, where $G_{8,9}$ are equivalent to G_∞. Crucially, it is the inability of agent

[2]Before an observation-action pair is experienced (i.e., when $n_{s,a} = 0$) the transition model is initialized to the identity function: $\hat{T}(s'|s,a) = 1$ iff $s' = s$.

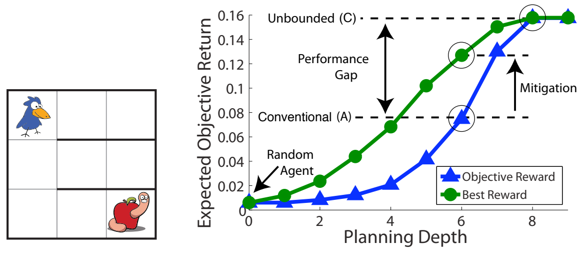


Figure 1. (a) Foraging domain used in Experiments 1 and 2. (b) Results from Experiment 1 on limited-depth planning, showing performance gains from using optimal internal rewards as a function of planning depth bound.

$G_{d<8}$ to encounter a confounded reward during planning from some states that we wish to mitigate via internal rewards.

Internal reward space. The general form of the rewards in all the experiments is $R_I(s,a,s') = \beta^{\mathsf{T}} \phi(s,a,s',h,G)$, where β is a parameter vector, and ϕ is a vector of features that may depend on states s and s', the action a, features of history h, and in some cases on internal variables specific to the agent G.

Our choice of features for this domain is driven by the following intuition. If an agent $G_{d<8}$ is more than d steps away from the worm, what action should it take? The agent could take random actions to explore randomly to achieve a state that is within d steps from the worm, but this will be inefficient. A good reward function would lead to some kind of systematic and persistent exploration and would thus be far more efficient. Specifically, we consider the reward space $R_I(s,a) = \beta_{\mathcal{E}} \phi_{\mathcal{E}}(s) + \beta_c \phi_c(s,a,h)$, where $\beta_{\mathcal{E}}$ and β_c are the two parameters, feature $\phi_{\mathcal{E}}(s)$ is 1 when the agent is satiated in state s and 0 otherwise, and feature $\phi_c(s,a,h) = 1 - \frac{1}{c(s,a,h)}$, where $c(s,a,h)$ is the number of time steps since the agent previously executed action a in state s within history h. Feature ϕ_c captures recency; the feature's value is high when the agent has not taken the indicated state-action pair recently. When β_c is positive, the agent is rewarded for taking actions that it has not taken recently from the current state. Note that when $\beta_{\mathcal{E}} = 1$ and $\beta_c = 0$, the internal reward is the confounded reward function.

Results. At each depth $d \in \{0, 1, \cdots, 9\}$, we separately optimized the internal reward function as measured by the mean objective return obtained during a 10,000 step horizon. We then evaluated the confounded reward function $R_{\mathcal{E}}$ and the approximate best internal reward $\hat{R}^*_{I\,d}$ at each depth d for 200,000 steps, averaged over 200 trials, to estimate the expected asymptotic objective return of each reward function.

As can be seen in Figure 1(b), the internal reward func-

Table 2. Results from Experiment 2 on the Bounded-State Agent in the Foraging Environment.

Agent type	$\beta_{\mathcal{E}}$	β_c	$\mathbb{E}[R_{\mathcal{E}}]$ per step
Random	0	0	0.0060 ± 2.46e-5
Conventional; $F_{R_{\mathcal{E}}}$	1	0	8.6e-6 ± 4.55e-7
F-Optimal; $F_{\hat{R}_I^*}$	0.147	0.989	0.0745 ± 2.15e-4
Unbounded	N/A	N/A	0.1153 ± 2.90e-5

tion $\hat{R}^*_{Id}$ performs at least as well as the objective reward function at all planning depths. When the planning depth is 0, both agents do not plan and simply act randomly. When the planning depth is 8 or more, the agent is in effect unbounded for this environment and the confounded reward acts optimally and no internal reward can do better. Between these two extremes, however, the use of internal reward greatly benefits the agent, and in some cases, $G(F_{\hat{R}^*_{Id}}, d)$ performs as well or better than $G(F_{R_{\mathcal{E}}}, d+2)$. These results are consistent with a general pattern that might be expected to hold across other kinds of agent bounds and environments: the benefit of internal reward reaches its maximum at some intermediate level of boundedness, but approaches zero as the agent either approaches the unbounded agent at one extreme, or a degenerate (perhaps random) agent at the other extreme.

3.2. Experiment 2: Mitigating bounds on state representation

The objective here is to show that internal rewards based on the same recency features as used in Experiment 1 can mitigate a different kind of agent-limitation, that of impoverished state representation.

The environment and designer's goals. Identical to Experiment 1, with one exception. In contrast to Experiment 1, the agent observes only its location, whether or not it is hungry, and whether or not it is colocated with the worm; the agent *cannot* see the worm unless it is colocated with it.

The agent and its bounds. The agent is an approximately unbounded-depth planning model-based learning agent G_∞. The decision process faced by the agent is not an MDP. Nevertheless, the agent uses the same model-learning procedure as was used in the previous experiment (the agent continually updates an estimated MDP model as if observations were Markov state). Given that the agent cannot observe the location of food, it cannot plan even with its infinite-look-ahead to go to it in the shortest path. Also, given that the agent's observations do not tell it what locations it has visited since the last time it ate food (and hence should be known to be empty locations), it cannot plan to avoid exploring locations that should be known not to have food based on the agent's history.

The reward space. We use the recency reward space, as described above in Experiment 1.

Results. In Table 2, we compare the agent G_∞ using the confounded return function $F_{R_{\mathcal{E}}}$ with the agent using the best internal return $F_{\hat{R}_I^*}$. Each estimated value is the mean objective utility obtained per step over $200,000$ steps, averaged over 200 trials. This table also shows the specific values of reward parameters; noteworthy is the relatively large coefficient for the recency feature relative to the coefficient for the confounded-reward feature. Additionally, we compare against two reference agents: a random agent, and an unbounded belief-state agent. The unbounded belief-state agent learns a POMDP model that helps the agent keep track of which potential food locations it has visited since the most recent time it ate, and it accounts for changes in belief during planning. As can be seen, the best internal reward performs much better than the objective reward, much better than a random agent, and not quite as well as the unbounded belief-state agent. The internal reward manages to achieve this despite being coupled with a model that is wholly inadequate at predicting the food location.

3.3. Experiment 3: Mitigating bounds on model representation

The objective here is to introduce local model accuracy, another general reward feature, and show that it can mitigate boundedness arising when an agent cannot model its environment uniformly accurately.

The environment and designer's goals. Figure 2(b) illustrates the Dark Room environment. After the worm is eaten, a new worm appears in the other right corner. Unlike Experiments 1 and 2, the agent's movement is stochastic—each movement action fails with probability 0.1, resulting in movement in a random direction. The special feature of this world is the dark room in the center spanning two locations. The agent's location sensor cannot distinguish the two locations inside the dark room but works perfectly outside the dark room. The agent always perceives the location of the food and whether it is hungry.

The agent and its bounds. The agent is an infinite-depth planning model-based learning agent (G_∞). The agent assumes a *factored* model of the three state variables—Agent Location, Food Location, and Hungry. Figure 2(a) depicts a graphical model of the factored assumptions that were designed to compactly but accurately capture the structure in the original

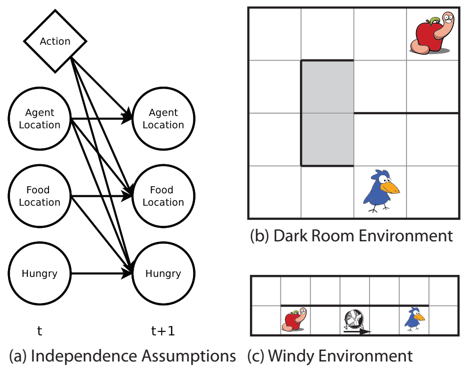


Figure 2. Model and Environments in Experiments 3 & 4.

fully-observable 3-Corridor foraging environment of experiments 1 & 2. However, the Dark Room environment violates the agent's Markov assumption for the location variable—in the dark room, the agent's next location observation is not independent of the history given its current location observation. The agent learns the probabilities of each component of the factored model using empirical counts, as was done for the unstructured modeling agents above.

The reward space. We include in the reward feature space two new environment-independent features. The first is a feature that estimates the *quality* of the agent's model as a function of state. The intuition is that such a feature might be exploited by the optimal reward function to motivate the agent to avoid areas it has difficulty modeling.

The specific model-error reward feature we formulate, ϕ_δ, keeps a simple moving average of the recent errors in each component model. For a correct model, the expected value of the parameters[3] in our model-error feature will be zero. However, they will tend to fluctuate around 0 in a stochastic environment. It is for this reason that the foraging environment has been modified to include stochastic movement—to show that the reward can distinguish between stochastic dynamics and an erroneous model.

One challenge in designing a model-building agent which is motivated to avoid states in which it has a bad model is that it will start with a bad model. To motivate the agent to build its model in the first place, we provide an additional reward feature $\phi_n(s,a)$ which is inversely proportional to how often action a has been taken in state s[4]. A positive coefficient for this feature would encourage the agent to experience state-action pairs it hasn't experienced often. The feature is by definition transient and its magnitude decreases steadily with increased experience.

In summary, the reward space we provide to the meta-optimization procedure is defined by $R_I(s,a,s') = \beta_\mathcal{E}\phi_\mathcal{E}(s) + \beta_\delta\phi_\delta(s,a,s') + \beta_n\phi_n(s,a)$.

Results. Table 3 presents the performance of the factored-model agent in the Dark Room environment (and Windy, described next). Each estimated value is the mean objective utility obtained per step over 200,000 steps, averaged over 200 trials. We tested the factored agent with the following set of rewards: $R_I \overset{\text{def}}{=} 0$ (this corresponds to random behavior), the objective reward $R_\mathcal{E}$, and the estimate of the optimal internal reward $\hat{R}_I^*$. In addition, we present as an approximation of the optimal agent the performance of the best agent which avoids the difficult-to-model state(s)[5].

The best internal reward significantly outperforms the agent with confounded reward. The latter is unable to learn to navigate up or down in the dark room; instead it repeatedly enters and exits the dark room on the same side, expecting to sometimes appear on the other side. The best internal rewards motivate behavior that avoids the room, instead going around the long way.

3.4. Experiment 4: Mitigating bounds on model representation, redux

The best internal reward in Experiment 3 is the result of a careful tuning (optimization) of the reward coefficients $\beta_\mathcal{E}$, β_δ, and β_n. Although the sign of these coefficients (positive, negative, and positive, respectively) is consistent with our expectations, their precise values are not something we had sharp intuitions about, and they may be determined largely by very specific

[3] Mathematically, let s_i denote the value of state feature i in state s, and let $\pi(i)$ denote the set features that feature i depends on in the previous state. We will refer to $\pi(i)$ as the parents of i. The agent's transition model is $\hat{T}(s'|s,a) = \prod_i \hat{T}_i(s'_i|s_{\pi(i)},a)$. For the model-error feature, we keep a parameter $\mu_{s_{\pi(i)},a,s'_i}$ for each unique $\langle s_{\pi(i)},a,s'_i\rangle$ tuple. After observing a transition from state s to state s', each component model has instantaneous error $\delta = 1-\hat{T}_i(s''_i|s_{\pi(i)},a)$ if $s''_i = s'_i$ and $\delta = -\hat{T}_i(s''_i|s_{\pi(i)},a)$ if $s''_i \neq s'_i$. Each error parameter matching the conditioned state and action features is then updated according to $\mu \leftarrow \mu + 0.1(\delta-\mu)$, where each μ is initialized to 0. These errors are then averaged across component models: $\phi_\delta(s,a,s') = \frac{1}{3}\sum_i(\mu_{i,s_{\pi(i)},a,s'_i})^2$.

[4] Specifically, we use the reward feature $\phi_n(s,a) = \frac{1}{3}\sum_i \frac{1}{n_{s_{\pi(i)},a}+1}$, where $n_{s_{\pi(i)},a}$ is the number of times action a was taken when the features in the parent set of i matched $s_{\pi(i)}$. See footnote 3 for the definition of π.

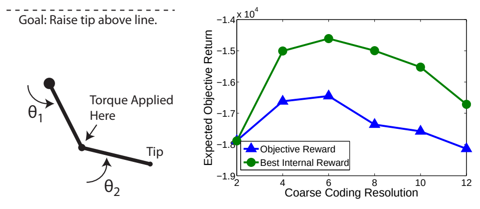
[5] In the Dark Room domain, the unbounded agent is the belief-state agent, but we do not present that agent's performance here.

Table 3. Factored Model Agent in Dark Room and Windy Environments (Experiments 3 & 4).

	Reward parameters			Dark Room	Windy
Agent Type	$\beta_{\mathcal{E}}$	β_{δ}	β_{n}	*mean* $R_{\mathcal{E}}$ *per step*	*mean* $R_{\mathcal{E}}$ *per step*
Random	0	0	0	0.0044 $\pm$ 1.67e-5	0.0018 $\pm$ 3.03e-5
Conventional, $F_{R_{\mathcal{E}}}$	1	0	0	0.0138 $\pm$ 2.43e-3	0.0335 $\pm$ 4.56e-3
F-Optimal (Dark Room), $F_{\hat{R}_I^*}$	0.267	-0.963	0.047	0.0895 $\pm$ 7.82e-5	0.0782 $\pm$ 3.18e-4
F-Optimal (Windy), $F_{\hat{R}_I^*}$	0.219	-0.975	0.048	0.0896 $\pm$ 4.38e-5	0.0806 $\pm$ 2.15e-4
Rounded, F_{R_I}	0.25	-1	0.05	0.0896 $\pm$ 6.89e-5	0.0801 $\pm$ 1.73e-4
Optimal	N/A	N/A	N/A	0.0903 $\pm$ 1.23e-5	0.0826 $\pm$ 9.34e-6

properties of the domain. It is therefore interesting to begin exploring how effective such specific rewards are when transferred to related but distinct domains, and this is one of the key objectives of Experiment 4.

The environment. Figure 2(c) illustrates the Windy environment. It shares the basic properties of the Dark Room environment—stochastic movement, unique location identifiers. The food source is in one of two locations, either the location pictured, or the location currently occupied by the agent. But instead of a dark room, this world has a different twist—there is a giant fan in the bottom-middle location. At all times, the fan directs a forceful wind in the direction *away* from the worm. If the agent is in the fan's location, its action will fail with probability 0.95 and the agent will transition to the state in the direction opposite from the worm. Otherwise, the attempted movement proceeds as normal, with another 0.10 chance of failing as in other locations.



(a) Acrobot (b) Acrobot Performance

Figure 3. Acrobot, Experiment 5.

The agent and its bounds. The agent is identical to that in Experiment 3. The Dark Room environment violated the Markov assumption in the factored model for the location variable but all other independence assumptions were accurate. The Windy environment, on the other hand, violates the independence assumption that the agent's next location is independent of the current location of the worm—when co-located with the fan, the next location is correlated with the worm's location. The factored agent with the objective reward attempts to push its way through the fan, though it is more efficient to walk around the long way.

The reward space. Identical to Experiment 3.

Results. Table 3 presents the results for the Windy environment. As in Dark Room, the agent with the best internal reward for Windy significantly outperforms the conventional agent. Table 3 also shows the performance of the agent in the Dark Room world when provided with the best *Windy* world reward, and vice versa. We also tested a single reward function derived by simply rounding off the β-coefficient values found optimal in both the Dark Room and Windy environments. All of the internal rewards, including the rounded reward, perform well in both environments, significantly outperforming the conventional agent and nearly closing the gap to the optimal agent (that takes the long way around). This is a small and preliminary illustration that the practice of reward design based on domain-independent features may yield reward functions that are robust across different environments.

3.5. Experiment 5: Mitigating bounds on function approximation and learning

The objective of this experiment is to extend our empirical work to a domain not constructed by us, and to learning agents with function approximation.

The environment and designer's goals. The Acrobot environment shown in Figure 3(a) is a popular research task, because it requires a non-trivial control policy and has a continuous state space. The fully-observable state space is 4 dimensional, with two joint angles θ_1 and θ_2, and two joint velocities $\dot{\theta}_1$ and $\dot{\theta}_2$. The agent's control is limited to applying torque to the indicated joint.

The version of Acrobot we use is exactly as specified by Sutton and Barto (1998). The task is episodic, and the goal is to minimize the number of time steps it takes the agent to raise the tip above a line equivalent to one arm's length above the fixed joint. Because we are interested in learning performance, we use an objective return function which negatively counts the number of time steps required to complete 50 episodes.

The agent and its bounds. The agent we use is a

Q-learning agent which approximates the Q-function using the coarse coding specified in Sutton and Barto (1998), with one slight modification. The referenced scheme uses 48 layers, each slicing the space along different subsets of dimensions. Each dimension is cut into k bins. Whereas the referenced implementation uses $k = 6$ for the angle features and $k = 7$ for the velocities, we used the same k in all dimensions and repeated our experiment for various values of k. We used a learning rate of $\alpha = 0.1/48$ and initialized the Q-function to 0. The discount factor is $\gamma = 1$.

This agent is bounded in two important ways. First, it is unlikely that Q-learning is an optimal learning algorithm. Second, the agent is bounded by its choice of function approximation. Like the depth limited planning case, the agent designer will be constrained in this choice by computational resource limits. Unlike the depth-limited planning agent, however, the optimal choice of resolution k is not clear—smaller values of k results in increased generalization, but can result in more errors induced by the approximation.

The reward space. The reward space we use is of the form $\beta_{\mathcal{E}} + \beta_h h$ where $h \in [-1, 1]$ is the height of the tip, and the goal height is $h = 0.5$. Note that setting $\langle \beta_{\mathcal{E}} = -1, \beta_h = 0 \rangle$ implements the objective reward function from Sutton and Barto (1998), while $\langle \beta_{\mathcal{E}} = .5, \beta_h = .5 \rangle$ effectively implements the objective reward function from Weaver and Tao (2001).

Results. The results are pictured in Figure 3(b). Except for the resolution $k = 2$, where the value function is too impoverished to take advantage of the height information, the internal reward helps at all resolutions. The best internal reward found was approximately $R_I(s, a) = -0.1 + h$ at all resolutions. It is clear from the graph that, starting from many initial resolutions θ, it can be more beneficial to optimize the reward function R_I than to further optimize θ.

4. Discussion and Future Directions

Future work should develop strong theoretical results justifying internal reward features which match agent limitations with environment properties. A recent line of research in RL has provided a few results that can be interpreted as providing examples, e.g., Kolter & Ng (2009) showed that adding reward inversely proportional to how often a state-action pair has been visited leads to performance that catches up in polynomial time to an unbounded (Bayes-optimal) agent. This paper provides a broader and clearer context by explicitly motivating the use of internal reward as mitigating agent boundedness. As one example of the import of the broader context, we note that Kolter & Ng's (and others, e.g., Strehl & Littman (2005)) results as well as the results on reward shaping (Ng et al., 1999) focus on achieving fast convergence to asymptotic behavior. Our experiments in foraging environments with consumable rewards that are renewed at unknown locations showed forms of agent boundedness for which it is optimal to explore persistently and never converge. Another consequence of the broad context of our claim of internal rewards mitigating agent boundedness is that it opens up the possibility (illustrated here empirically) of characterizing different fundamental forms of boundedness and corresponding general reward features that allow for mitigation by internal rewards. Finally, while we focused on artificial agents in this paper, some of the implications of internal rewards for natural agents where evolution plays the role of agent-designer were explored in Singh et. al. (2009).

Acknowledgments: This work was supported by the Air Force Office of Scientific Research under grant FA9550-08-1-0418 as well as by NSF grant IIS 0905146. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

Kolter, J. Zico and Ng, Andrew Y. Near-Bayesian Exploration in Polynomial Time. *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1–8, 2009.

Ng, Andrew Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International conference on Machine Learning*, pp. 278–287, Bled, Slovenia, 1999.

Singh, Satinder, Lewis, Richard L., and Barto, Andrew G. Where Do Rewards Come From? In *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 2601–2606, 2009.

Strehl, Alexander L. and Littman, Michael L. A theoretical analysis of model-based interval estimation. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, pp. 857–864, 2005.

Sutton, Richard S. and Barto, Andrew G. *Reinforcement Learning: An Introduction.* The MIT Press, 1998.

Weaver, Lex and Tao, Nigel. The Optimal Reward Baseline for Gradient-Based Reinforcement Learning. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001.