

# Near Optimal Behavior via Approximate State Abstraction

David Abel<sup>†</sup>  
Brown University  
david\_abel@brown.edu

D. Ellis Hershkowitz<sup>†</sup>  
Carnegie Mellon University  
dershko@cs.cmu.edu

Michael L. Littman  
Brown University  
mlittman@cs.brown.edu

## Abstract

The combinatorial explosion that plagues planning and reinforcement learning (RL) algorithms can be moderated using state abstraction. Prohibitively large task representations can be condensed such that essential information is preserved, and consequently, solutions are tractably computable. However, exact abstractions, which treat only fully-identical situations as equivalent, fail to present opportunities for abstraction in environments where no two situations are exactly alike. In this work, we investigate approximate state abstractions, which treat nearly-identical situations as equivalent. We present theoretical guarantees of the quality of behaviors derived from four types of approximate abstractions. Additionally, we empirically demonstrate that approximate abstractions lead to reduction in task complexity and bounded loss of optimality of behavior in a variety of environments.

## 1 Introduction

Abstraction plays a fundamental role in learning. Through abstraction, intelligent agents may reason about only the salient features of their environment while ignoring what is irrelevant. Consequently, agents are able to solve considerably more complex problems than they would be able to without the use of abstraction. However, *exact abstractions*, which treat only fully-identical situations as equivalent, require complete knowledge that is computationally intractable to obtain. Furthermore, often no two situations are identical, so exact abstractions are often ineffective. To overcome these issues, we investigate *approximate abstractions* that enable agents to treat sufficiently similar situations as identical. This work characterizes the impact of equating “sufficiently similar” states in the context of planning and RL in Markov Decision Processes (MDPs). The remainder of our introduction contextualizes these intuitions in MDPs.

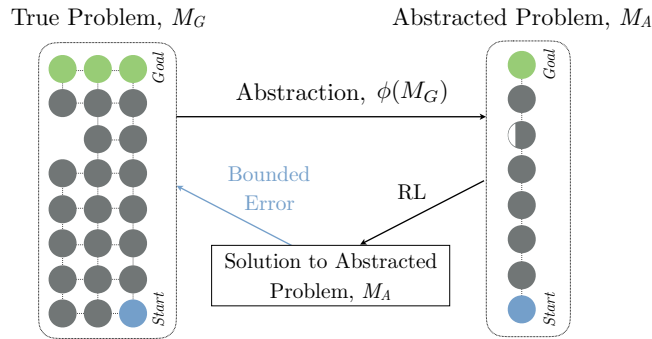


Figure 1: We investigate families of approximate state abstraction functions that induce abstract MDP’s whose optimal policies have bounded value in the original MDP.

A previous version of this paper was published in the Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

<sup>†</sup>The first two authors contributed equally.

Solving for optimal behavior in MDPs in a planning setting is known to be P-Complete in the size of the state space [28, 25]. Similarly, many RL algorithms for solving MDPs are known to require a number of samples polynomial in the size of the state space [31]. Although polynomial runtime or sample complexity may seem like a reasonable constraint, the size of the state space of an MDP grows super-polynomially with the number of variables that characterize the domain - a result of Bellman’s curse of dimensionality. Thus, solutions polynomial in state space size are often ineffective for sufficiently complex tasks. For instance, a robot involved in a pick-and-place task might be able to employ planning algorithms to solve for how to manipulate some objects into a desired configuration in time polynomial in the number of states, but the number of states it must consider grows exponentially with the number of objects with which it is working [1].

Thus, a key research agenda for planning and RL is leveraging abstraction to reduce large state spaces [2, 21, 10, 12, 6]. This agenda has given rise to methods that reduce *ground* MDPs with large state spaces to *abstract* MDPs with smaller state spaces by aggregating states according to some notion of equality or similarity. In the context of MDPs, we understand exact abstractions as those that aggregate states with equal values of particular quantities, for example, optimal  $Q$ -values. Existing work has characterized how exact abstractions can fully maintain optimality in MDPs [24, 8].

The thesis of this work is that performing approximate abstraction in MDPs by relaxing the state aggregation criteria from equality to similarity achieves polynomially bounded error in the resulting behavior while offering three benefits. First, approximate abstractions employ the sort of knowledge that we expect a planning or learning algorithm to compute without fully solving the MDP. In contrast, exact abstractions often require solving for optimal behavior, thereby defeating the purpose of abstraction. Second, because of their relaxed criteria, approximate abstractions can achieve greater degrees of compression than exact abstractions. This difference is particularly important in environments where no two states are identical. Third, because the state aggregation criteria are relaxed to near equality, approximate abstractions are able to tune the aggressiveness of abstraction by adjusting what they consider sufficiently similar states.

We support this thesis by describing four different types of approximate abstraction functions that preserve near-optimal behavior by aggregating states on different criteria:  $\tilde{\phi}_{Q^*,\varepsilon}$ , on similar optimal  $Q$ -values,  $\tilde{\phi}_{\text{model},\varepsilon}$ , on similarity of rewards and transitions,  $\tilde{\phi}_{\text{bolt},\varepsilon}$ , on similarity of a Boltzmann distribution over optimal  $Q$ -values, and  $\tilde{\phi}_{\text{mult},\varepsilon}$ , on similarity of a multinomial distribution over optimal  $Q$ -values. Furthermore, we empirically demonstrate the relationship between the degree of compression and error incurred on a variety of MDPs.

This paper is organized as follows. In the next section, we introduce the necessary terminology and background of MDPs and state abstraction. Section 3 surveys existing work on state abstraction applied to sequential decision making. Section 5 introduces our primary result; bounds on the error guaranteed by four classes of approximate state abstraction. The following two sections introduce simulated domains used in experiments (Section 6), and a discussion of experiments in which we apply one class of approximate abstraction to a variety of different tasks to empirically illustrate the relationship between degree of compression and error incurred (Section 7).

## 2 MDPs and Sequential Decision Making

An MDP is a problem representation for sequential decision making agents, represented by a five-tuple:  $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ . Here,  $\mathcal{S}$  is a finite state space;  $\mathcal{A}$  is a finite set of actions available to the agent;  $\mathcal{T}$  denotes  $\mathcal{T}(s, a, s')$ , the probability of an agent transitioning to state  $s' \in \mathcal{S}$  after applying action  $a \in \mathcal{A}$  in state  $s \in \mathcal{S}$ ;  $\mathcal{R}(s, a)$  denotes the reward received by the agent for executing action  $a$  in state  $s$ ;  $\gamma \in [0, 1]$  is a discount factor that determines how much the agent prefers future rewards over immediate rewards. We assume without loss of generality that the range of all reward functions is normalized to  $[0, 1]$ . The solution to an MDP is called a policy, denoted  $\pi : \mathcal{S} \mapsto \mathcal{A}$ .

The objective of an agent is to solve for the policy that maximizes its expected discounted reward from any state, denoted  $\pi^*$ . We denote the expected discounted reward for following policy  $\pi$  from state  $s$  as the value of the state under that policy,  $V^\pi(s)$ . We similarly denote the expected discounted reward for taking

action  $a \in \mathcal{A}$  and then following policy  $\pi$  from state  $s$  forever after as  $Q^\pi(s, a)$ , defined by the Bellman Equation as:

$$Q^\pi(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s'} \mathcal{T}(s, a, s') Q^\pi(s', \pi(s')). \quad (1)$$

We let RMAX denote the maximum reward (which is 1), and QMAX denote the maximum  $Q$  value, which is  $\frac{\text{RMAX}}{1-\gamma}$ . The value function,  $V$ , defined under a given policy, denoted  $V^\pi(s)$ , is defined as:

$$V^\pi(s) = Q^\pi(s, \pi(s)). \quad (2)$$

Lastly, we denote the value and  $Q$  functions under the optimal policy as  $V^*$  or  $V^{\pi^*}$  and  $Q^*$  or  $Q^{\pi^*}$ , respectively. For further background, see Kaelbling et al. [22].

## 3 Related Work

Several other projects have addressed similar topics.

### 3.1 Approximate State Abstraction

Dean et al. [9] leverage the notion of *bisimulation* to investigate partitioning an MDP’s state space into clusters of states whose transition model and reward function are within  $\varepsilon$  of each other. They develop an algorithm called Interval Value Iteration (IVI) that converges to the correct bounds on a family of abstract MDPs called Bounded MDPs.

Several approaches build on Dean et al. [9]. Ferns et al. [14, 15] investigated state similarity metrics for MDPs; they bounded the value difference of ground states and abstract states for several bisimulation metrics that induce an abstract MDP. This differs from our work which develops a theory of abstraction that bounds the suboptimality of applying the optimal policy of an abstract MDP to its ground MDP, covering four types of state abstraction, one of which closely parallels bisimulation. Even-Dar and Mansour [13] analyzed different distance metrics used in identifying state space partitions subject to  $\varepsilon$ -similarity, also providing value bounds (their Lemma 4) for  $\varepsilon$ -homogeneity subject to the  $L_\infty$  norm, which parallels our Claim 2. Ortner [27] developed an algorithm for learning partitions in an online setting by taking advantage of the confidence bounds for  $\mathcal{T}$  and  $\mathcal{R}$  provided by UCRL [3].

Hutter [18, 17] investigates state aggregation beyond the MDP setting. Hutter presents a variety of results for aggregation functions in reinforcement learning. Most relevant to our investigation is Hutter’s Theorem 8, which illustrates properties of aggregating states based on similar  $Q$  values. Hutter’s Theorem part (a) parallels our Claim: both bound the value difference between ground and abstraction states, and part (b) is analogous to our Lemma 1: both bound the value difference of applying the optimal abstraction policy in the ground, and part (c) is a repetition of the comment given by Li et al. [24] that  $Q^*$  abstractions preserve the optimal value function. For Lemma 1, our proof strategies differ from Hutter’s, but the result is the same.

Approximate state abstraction has also been applied to the planning problem, in which the agent is given a model of its environment and must compute a plan that satisfies some goal. Hostetler et al. [16] apply state abstraction to Monte Carlo Tree Search and expectimax search, giving value bounds of applying the optimal abstract action in the ground tree(s), similarly to our setting. Dearden and Boutilier [10] also formalize state-abstraction for planning, focusing on abstractions that are quickly computed and offer bounded value. Their primary analysis is on abstractions that remove negligible literals from the planning domain description, yielding value bounds for these abstractions and a means of incrementally improving abstract solutions to planning problems. Jiang et al. [20] analyze a similar setting, applying abstractions to the Upper Confidence Bound applied to Trees algorithm adapted for planning, introduced by Kocsis and Szepesvári [23].

Mandel et al. [26] advance Bayesian aggregation in RL to define Thompson Clustering for Reinforcement Learning (TCRL), an extension of which achieves near-optimal Bayesian regret bounds. Jiang [19] analyze the problem of choosing between two candidate abstractions. They develop an algorithm based on statistical

tests that trades of the approximation error with the estimation error of the two abstractions, yielding a loss bound on the quality of the chosen policy.

### 3.2 Specific Abstraction Algorithms

Many previous works have targeted the creation of algorithms that enable state abstraction for MDPs. Andre and Russell [2] investigated a method for state abstraction in hierarchical reinforcement learning leveraging a programming language called ALISP that promotes the notion of *safe* state abstraction. Agents programmed using ALISP can ignore irrelevant parts of the state, achieving abstractions that maintain optimality. Dietterich [12] developed MAXQ, a framework for composing tasks into an abstracted hierarchy where state aggregation can be applied. Bakker and Schmidhuber [4] also target hierarchical abstraction, focusing on subgoal discovery. Jong and Stone [21] introduced a method called *policy-irrelevance* in which agents identify (online) which state variables may be safely abstracted away in a factored-state MDP. Dayan and Hinton [7] develop “Feudal Reinforcement Learning” which presents an early form of hierarchical RL that restructures  $Q$ -Learning to manage the decomposition of a task into subtasks. For a more complete survey of algorithms that leverage state abstraction in past reinforcement-learning papers, see Li et al. [24], and for a survey of early works on hierarchical reinforcement learning, see Barto and Mahadevan [5].

### 3.3 Exact Abstraction Framework

Li et al. [24] developed a framework for exact state abstraction in MDPs. In particular, the authors defined five types of state aggregation functions, inspired by existing methods for state aggregation in MDPs. We generalize two of these five types,  $\phi_{Q^*}$  and  $\phi_{\text{model}}$ , to the approximate abstraction case. Our generalizations are equivalent to theirs when exact criteria are used (i.e.  $\varepsilon = 0$ ). Additionally, when exact criteria are used our bounds indicate that no value is lost, which is one of core results of Li et al. [24]. Walsh et al. [34] build on the framework they previously developed by showing empirically how to transfer abstractions between structurally related MDPs.

## 4 Abstraction Notation

We build upon the notation used by Li et al. [24], who introduced a unifying theoretical framework for state abstraction in MDPs.

**Definition 1** ( $M_G, M_A$ ): *We understand an abstraction as a mapping from the state space of a ground MDP,  $M_G$ , to that of an abstract MDP,  $M_A$ , using a state aggregation scheme. Consequently, this mapping induces an abstract MDP. Let  $M_G = \langle \mathcal{S}_G, \mathcal{A}, \mathcal{T}_G, \mathcal{R}_G, \gamma \rangle$  and  $M_A = \langle \mathcal{S}_A, \mathcal{A}, \mathcal{T}_A, \mathcal{R}_A, \gamma \rangle$ .*

**Definition 2** ( $\mathcal{S}_A, \phi$ ): *The states in the abstract MDP are constructed by applying a state aggregation function,  $\phi$ , to the states in the ground MDP,  $\mathcal{S}_A$ . More specifically,  $\phi$  maps a state in the ground MDP to a state in the abstract MDP:*

$$\mathcal{S}_A = \{\phi(s) \mid s \in \mathcal{S}_G\}. \tag{3}$$

**Definition 3** ( $G$ ): *Given a  $\phi$ , each ground state has associated with it the ground states with which it is aggregated. Similarly, each abstract state has its constituent ground states. We let  $G$  be the function that retrieves these states:*

$$G(s) = \begin{cases} \{g \in \mathcal{S}_G \mid \phi(g) = \phi(s)\}, & \text{if } s \in \mathcal{S}_G, \\ \{g \in \mathcal{S}_G \mid \phi(g) = s\}, & \text{if } s \in \mathcal{S}_A. \end{cases} \tag{4}$$

The abstract reward function and abstract transition dynamics for each abstract state are a weighted combination of the rewards and transitions for each ground state in the abstract state.

**Definition 4 ( $\omega(s)$ ):** We refer to the weight associated with a ground state,  $s \in \mathcal{S}_G$  by  $\omega(s)$ . The only restriction placed on the weighting scheme is that it induces a probability distribution on the ground states of each abstract state:

$$\forall s \in \mathcal{S}_G \left( \sum_{s \in G(s)} \omega(s) \right) = 1 \quad \text{AND} \quad \omega(s) \in [0, 1]. \quad (5)$$

**Definition 5 ( $\mathcal{R}_A$ ):** The abstract reward function  $\mathcal{R}_A : \mathcal{S}_A \times \mathcal{A} \mapsto [0, 1]$  is a weighted sum of the rewards of each of the ground states that map to the same abstract state:

$$\mathcal{R}_A(s, a) = \sum_{g \in G(s)} \mathcal{R}_G(g, a) \omega(g). \quad (6)$$

**Definition 6 ( $\mathcal{T}_A$ ):** The abstract transition function  $\mathcal{T}_A : \mathcal{S}_A \times \mathcal{A} \times \mathcal{S}_A \mapsto [0, 1]$  is a weighted sum of the transitions of each of the ground states that map to the same abstract state:

$$\mathcal{T}_A(s, a, s') = \sum_{g \in G(s)} \sum_{g' \in G(s')} \mathcal{T}_G(g, a, g') \omega(g). \quad (7)$$

## 5 Approximate State Abstraction

Here, we introduce our formal analysis of approximate state abstraction, including results bounding the error associated with these abstraction methods. In particular, we demonstrate that abstractions based on approximate  $Q^*$  similarity (5.2), approximate model similarity (5.3), and approximate similarity between distributions over  $Q^*$ , for both Boltzmann (5.4) and multinomial (5.5) distributions induce abstract MDPs for which the optimal policy has bounded error in the ground MDP.

We first introduce some additional notation.

**Definition 7 ( $\pi_A^*$ ,  $\pi_G^*$ ):** We let  $\pi_A^* : \mathcal{S}_A \rightarrow \mathcal{A}$  and  $\pi_G^* : \mathcal{S}_G \rightarrow \mathcal{A}$  stand for the optimal policies in the abstract and ground MDPs, respectively.

We are interested in how the optimal policy in the abstract MDP performs in the ground MDP. As such, we formally define the policy in the ground MDP derived from optimal behavior in the abstract MDP:

**Definition 8 ( $\pi_{GA}$ ):** Given a state  $s \in \mathcal{S}_G$  and a state aggregation function,  $\phi$ ,

$$\pi_{GA}(s) = \pi_A^*(\phi(s)). \quad (8)$$

We now define types of abstraction based on functions of state–action pairs.

**Definition 9 ( $\tilde{\phi}_{f,\varepsilon}$ ):** Given a function  $f : \mathcal{S}_G \times \mathcal{A} \rightarrow \mathbb{R}$  and a fixed non-negative  $\varepsilon \in \mathbb{R}$ , we define  $\tilde{\phi}_{f,\varepsilon}$  as a type of approximate state aggregation function that satisfies the following for any two ground states  $s_1, s_2$ :

$$\tilde{\phi}_{f,\varepsilon}(s_1) = \tilde{\phi}_{f,\varepsilon}(s_2) \rightarrow \forall a |f(s_1, a) - f(s_2, a)| \leq \varepsilon. \quad (9)$$

That is, when  $\tilde{\phi}_{f,\varepsilon}$  aggregates states, all aggregated states have values of  $f$  within  $\varepsilon$  of each other for all actions.

Finally, we establish notation to distinguish between the *ground* and *abstract* value ( $V$ ) and action value ( $Q$ ) functions.

**Definition 10 ( $Q_G$ ,  $V_G$ ):** Let  $Q_G = Q^{\pi_G^*} : \mathcal{S}_G \times \mathcal{A} \rightarrow \mathbb{R}$  and  $V_G = V^{\pi_G^*} : \mathcal{S}_G \rightarrow \mathbb{R}$  denote the optimal  $Q$  and optimal value functions in the ground MDP.

**Definition 11 ( $Q_A$ ,  $V_A$ ):** Let  $Q_A = Q^{\pi_A^*} : \mathcal{S}_A \times \mathcal{A} \rightarrow \mathbb{R}$  and  $V_A = V^{\pi_A^*} : \mathcal{S}_A \rightarrow \mathbb{R}$  stand for the optimal  $Q$  and optimal value functions in the abstract MDP.

### 5.1 Main Result

We now introduce the main result of the paper.

**Theorem 1.** *There exist at least four types of approximate state aggregation functions,  $\tilde{\phi}_{Q^*,\varepsilon}$ ,  $\tilde{\phi}_{model,\varepsilon}$ ,  $\tilde{\phi}_{bolt,\varepsilon}$  and  $\tilde{\phi}_{mult,\varepsilon}$ , for which the optimal policy in the abstract MDP, applied to the ground MDP, has suboptimality bounded polynomially in  $\varepsilon$ :*

$$\forall_{s \in \mathcal{S}_G} V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq 2\varepsilon\eta_f \quad (10)$$

Where  $\eta_f$  differs between abstraction function families:

$$\begin{aligned} \eta_{Q^*} &= \frac{1}{(1-\gamma)^2} \\ \eta_{model} &= \frac{1 + \gamma(|\mathcal{S}_G| - 1)}{(1-\gamma)^3} \\ \eta_{bolt} &= \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + \varepsilon k_{bolt} + k_{bolt}\right)}{(1-\gamma)^2} \\ \eta_{mult} &= \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + k_{mult}\right)}{(1-\gamma)^2} \end{aligned}$$

For  $\eta_{bolt}$  and  $\eta_{mult}$ , we also assume that the difference in the normalizing terms of each distribution is bounded by some non-negative constant,  $k_{mult}, k_{bolt} \in \mathbb{R}$ , of  $\varepsilon$ :

$$\begin{aligned} \left| \sum_i Q_G(s_1, a_i) - \sum_j Q_G(s_2, a_j) \right| &\leq k_{mult} \times \varepsilon \\ \left| \sum_i e^{Q_G(s_1, a_i)} - \sum_j e^{Q_G(s_2, a_j)} \right| &\leq k_{bolt} \times \varepsilon \end{aligned}$$

Naturally, the value bound of Equation 10 is meaningless for  $2\varepsilon\eta_f \geq \frac{R_{MAX}}{1-\gamma} = \frac{1}{1-\gamma}$ , since this is the maximum possible value in any MDP (and we assumed the range of  $\mathcal{R}$  is  $[0, 1]$ ). In light of this, observe that for  $\varepsilon = 0$ , all of the above bounds are exactly 0. Any value of  $\varepsilon$  interpolated between these two points achieves different degrees of abstraction, with different degrees of bounded loss.

We now introduce each approximate aggregation family and prove the theorem by proving the specific value bound for each function type.

## 5.2 Optimal Q Function: $\tilde{\phi}_{Q^*,\varepsilon}$

We consider an approximate version of Li et al. [24]’s  $\phi_{Q^*}$ . In our abstraction, states are aggregated together when their optimal  $Q$ -values are within  $\varepsilon$ .

**Definition 12** ( $\tilde{\phi}_{Q^*,\varepsilon}$ ): *An approximate  $Q$  function abstraction has the same form as Equation 9:*

$$\tilde{\phi}_{Q^*,\varepsilon}(s_1) = \tilde{\phi}_{Q^*,\varepsilon}(s_2) \rightarrow \forall_a |Q_G(s_1, a) - Q_G(s_2, a)| \leq \varepsilon. \quad (11)$$

**Lemma 1.** *When a  $\tilde{\phi}_{Q^*,\varepsilon}$  type abstraction is used to create the abstract MDP:*

$$\forall_{s \in \mathcal{S}_G} V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq \frac{2\varepsilon}{(1-\gamma)^2}. \quad (12)$$

**Proof of Lemma 1:** We first demonstrate that  $Q$ -values in the abstract MDP are close to  $Q$ -values in the ground MDP (Claim 1). We next leverage Claim 1 to demonstrate that the optimal action in the abstract MDP is nearly optimal in the ground MDP (Claim 2). Lastly, we use Claim 2 to conclude Lemma 1 (Claim 3).

**Claim 1.** *Optimal  $Q$ -values in the abstract MDP closely resemble optimal  $Q$ -values in the ground MDP:*

$$\forall_{s_G \in \mathcal{S}_G, a} |Q_G(s_G, a) - Q_A(\tilde{\phi}_{Q^*, \varepsilon}(s_G), a)| \leq \frac{\varepsilon}{1 - \gamma}. \quad (13)$$

Consider a non-Markovian decision process of the same form as an MDP,  $M_T = \langle \mathcal{S}_T, \mathcal{A}_G, \mathcal{R}_T, \mathcal{T}_T, \gamma \rangle$ , parameterized by integer an  $T$ , such that for the first  $T$  time steps the reward function, transition dynamics and state space are those of the abstract MDP,  $M_A$ , and after  $T$  time steps the reward function, transition dynamics and state spaces are those of  $M_G$ . Thus,

$$\begin{aligned} \mathcal{S}_T &= \begin{cases} \mathcal{S}_G & \text{if } T = 0 \\ \mathcal{S}_A & \text{o/w} \end{cases} \\ \mathcal{R}_T(s, a) &= \begin{cases} \mathcal{R}_G(s, a) & \text{if } T = 0 \\ \mathcal{R}_A(s, a) & \text{o/w} \end{cases} \\ \mathcal{T}_T(s, a, s') &= \begin{cases} \mathcal{T}_G(s, a, s') & \text{if } T = 0 \\ \sum_{g \in G(s)} [\mathcal{T}_G(g, a, s') \omega(g)] & \text{if } T = 1 \\ \mathcal{T}_A(s, a, s') & \text{o/w} \end{cases} \end{aligned}$$

The  $Q$ -value of state  $s$  in  $\mathcal{S}_T$  for action  $a$  is:

$$Q_T(s, a) = \begin{cases} Q_G(s, a) & \text{if } T = 0 \\ \sum_{g \in G(s)} [Q_G(g, a) \omega(g)] & \text{if } T = 1 \\ \mathcal{R}_A(s, a) + \sigma_{T-1}(s, a) & \text{o/w} \end{cases} \quad (14)$$

where:

$$\sigma_{T-1}(s, a) = \gamma \sum_{s_{A'} \in \mathcal{S}_A} \mathcal{T}_A(s, a, s_{A'}) \max_{a'} Q_{T-1}(s_{A'}, a').$$

We proceed by induction on  $T$  to show that:

$$\forall_{T, s_G \in \mathcal{S}_G, a} |Q_T(s_T, a) - Q_G(s_G, a)| \leq \sum_{t=0}^{T-1} \varepsilon \gamma^t, \quad (15)$$

where  $s_T = s_G$  if  $T = 0$  and  $s_T = \tilde{\phi}_{Q^*, \varepsilon}(s_G)$  otherwise.

*Base Case:  $T = 0$*

When  $T = 0$ ,  $Q_T = Q_G$ , so this base case trivially follows.

*Base Case:  $T = 1$*

By definition of  $Q_T$ , we have that  $Q_1$  is

$$Q_1(s, a) = \sum_{g \in G(s)} [Q_G(g, a) \omega(g)].$$

Since all co-aggregated states have  $Q$ -values within  $\varepsilon$  of one another and  $\omega(g)$  induces a convex combination,

$$\begin{aligned} Q_1(s_T, a) &\leq \varepsilon \gamma^t + \varepsilon + Q_G(s_G, a) \\ \therefore |Q_1(s_T, a) - Q_G(s_G, a)| &\leq \sum_{t=0}^1 \varepsilon \gamma^t. \end{aligned}$$

*Inductive Case:  $T > 1$*

We assume as our inductive hypothesis that:

$$\forall s_G \in \mathcal{S}_G, a | Q_{T-1}(s_T, a) - Q_G(s_G, a) | \leq \sum_{t=0}^{T-2} \varepsilon \gamma^t.$$

Consider a fixed but arbitrary state,  $s_G \in \mathcal{S}_G$ , and fixed but arbitrary action  $a$ . Since  $T > 1$ ,  $s_T$  is  $\tilde{\phi}_{Q^*, \varepsilon}(s_G)$ . By definition of  $Q_T(s_T, a)$ ,  $\mathcal{R}_A$ ,  $\mathcal{T}_A$ :

$$Q_T(s_T, a) = \sum_{g \in G(s_T)} \omega(g) \times \left[ \mathcal{R}_G(g, a) + \gamma \sum_{g' \in \mathcal{S}_G} \mathcal{T}_G(g, a, g') \max_{a'} Q_{T-1}(g', a') \right].$$

Applying our inductive hypothesis yields:

$$Q_T(s_T, a) \leq \sum_{g \in G(s_T)} \omega(g) \times \left[ R_G(g, a) + \gamma \sum_{g' \in \mathcal{S}_G} T_G(g, a, g') \max_{a'} (Q_G(g', a') + \sum_{t=0}^{T-2} \varepsilon \gamma^t) \right].$$

Since all aggregated states have  $Q$ -values within  $\varepsilon$  of one another:

$$Q_T(s_T, a) \leq \gamma \sum_{t=0}^{T-2} \varepsilon \gamma^t + \varepsilon + Q_G(s_G, a).$$

Since  $s_G$  is arbitrary we conclude Equation 15. As  $T \rightarrow \infty$ ,  $\sum_{t=0}^{T-1} \varepsilon \gamma^t \rightarrow \frac{\varepsilon}{1-\gamma}$  by the sum of infinite geometric series and  $Q_T \rightarrow Q_A$ . Thus, Equation 15 yields Claim 1.

**Claim 2.** Consider a fixed but arbitrary state,  $s_G \in \mathcal{S}_G$  and its corresponding abstract state  $s_A = \tilde{\phi}_{Q^*, \varepsilon}(s_G)$ . Let  $a_G^*$  stand for the optimal action in  $s_G$ , and  $a_A^*$  stand for the optimal action in  $s_A$ :

$$a_G^* = \arg \max_a Q_G(s_G, a), \quad a_A^* = \arg \max_a Q_A(s_A, a).$$

The optimal action in the abstract MDP has a  $Q$ -value in the ground MDP that is nearly optimal:

$$V_G(s_G) \leq Q_G(s_G, a_A^*) + \frac{2\varepsilon}{1-\gamma}. \tag{16}$$

By Claim 1,

$$V_G(s_G) = Q_G(s_G, a_G^*) \leq Q_A(s_A, a_G^*) + \frac{\varepsilon}{1-\gamma}. \tag{17}$$

By the definition of  $a_A^*$ , we know that

$$Q_A(s_A, a_G^*) + \frac{\varepsilon}{1-\gamma} \leq Q_A(s_A, a_A^*) + \frac{\varepsilon}{1-\gamma}. \tag{18}$$

Lastly, again by Claim 1, we know

$$Q_A(s_A, a_A^*) + \frac{\varepsilon}{1-\gamma} \leq Q_G(s_G, a_A^*) + \frac{2\varepsilon}{1-\gamma}. \tag{19}$$

Therefore, Equation 16 follows.

**Claim 3.** Lemma 1 follows from Claim 2.



Consider the policy for  $M_G$  of following the optimal abstract policy  $\pi_A^*$  for  $t$  steps and then following the optimal ground policy  $\pi_G^*$  in  $M_G$ :

$$\pi_{A,t}(s) = \begin{cases} \pi_G^*(s) & \text{if } t = 0 \\ \pi_{GA}(s) & \text{if } t > 0 \end{cases} \quad (20)$$

For  $t > 0$ , the value of this policy for  $s_G \in \mathcal{S}_G$  in the ground MDP is:

$$V_G^{\pi_{A,t}}(s_G) = R_G(s, \pi_{A,t}(s_G)) + \gamma \sum_{s_G' \in \mathcal{S}_G} \mathcal{T}_G(s_G, a, s_G') V_G^{\pi_{A,t-1}}(s_G').$$

For  $t = 0$ ,  $V_G^{\pi_{A,t}}(s_G)$  is simply  $V_G(s_G)$ .

We now show by induction on  $t$  that

$$\forall t, s_G \in \mathcal{S}_g V_G(s_G) \leq V_G^{\pi_{A,t}}(s_G) + \sum_{i=0}^t \gamma^i \frac{2\varepsilon}{1-\gamma}. \quad (21)$$

*Base Case:  $t = 0$*

By definition, when  $t = 0$ ,  $V_G^{\pi_{A,t}} = V_G$ , so our bound trivially holds in this case.

*Inductive Case:  $t > 0$*

Consider a fixed but arbitrary state  $s_G \in \mathcal{S}_G$ . We assume for our inductive hypothesis that

$$V_G(s_G) \leq V_G^{\pi_{A,t-1}}(s_G) + \sum_{i=0}^{t-1} \gamma^i \frac{2\varepsilon}{1-\gamma}. \quad (22)$$

By definition,

$$V_G^{\pi_{A,t}}(s_G) = R_G(s, \pi_{A,t}(s_G)) + \gamma \sum_{g'} \mathcal{T}_G(s_G, a, s_G') V_G^{\pi_{A,t-1}}(s_G').$$

Applying our inductive hypothesis yields:

$$V_G^{\pi_{A,t}}(s_G) \geq R_G(s_G, \pi_{A,t}(s_G)) + \gamma \sum_{s_G'} \mathcal{T}_G(s_G, \pi_{A,t}(s_G), s_G') \left( V_G(s_G') - \sum_{i=0}^{t-1} \gamma^i \frac{2\varepsilon}{1-\gamma} \right).$$

Therefore,

$$V_G^{\pi_{A,t}}(s_G) \geq -\gamma \sum_{i=0}^{t-1} \gamma^i \frac{2\varepsilon}{1-\gamma} + Q_G(s_G, \pi_{A,t}(s_G)).$$

Applying Claim 2 yields:

$$\begin{aligned} V_G^{\pi_{A,t}}(s_G) &\geq -\gamma \sum_{i=0}^{t-1} \gamma^i \frac{2\varepsilon}{1-\gamma} - \frac{2\varepsilon}{1-\gamma} + V_G(s_G) \\ \therefore V_G(s_G) &\leq V_G^{\pi_{A,t}}(s_G) + \sum_{i=0}^t \gamma^i \frac{2\varepsilon}{1-\gamma}. \end{aligned}$$

Since  $s_G$  was arbitrary, we conclude that our bound holds for all states in  $\mathcal{S}_G$  for the inductive case. Thus, from our base case and induction, we conclude that

$$\forall t, s_G \in \mathcal{S}_g V_G^{\pi_G^*}(s_G) \leq V_G^{\pi_{A,t}}(s_G) + \sum_{i=0}^t \gamma^i \frac{2\varepsilon}{1-\gamma}. \quad (23)$$

Note that as  $t \rightarrow \infty$ ,  $\sum_{i=0}^t \gamma^i \frac{2\varepsilon}{1-\gamma} \rightarrow \frac{2\varepsilon}{(1-\gamma)^2}$  by the sum of infinite geometric series and  $\pi_{A,t}(s) \rightarrow \pi_{GA}$ . Thus, we conclude Lemma 1.  $\square$

### 5.3 Model Similarity: $\tilde{\phi}_{model,\varepsilon}$

Now, consider an approximate version of Li et al. [24]’s  $\phi_{model}$ , where states are aggregated together when their rewards and transitions are within  $\varepsilon$ .

**Definition 13** ( $\tilde{\phi}_{model,\varepsilon}$ ): We let  $\tilde{\phi}_{model,\varepsilon}$  define a type of abstraction that, for fixed  $\varepsilon$ , satisfies:

$$\tilde{\phi}_{model,\varepsilon}(s_1) = \tilde{\phi}_{model,\varepsilon}(s_2) \rightarrow \forall_a |\mathcal{R}_G(s_1, a) - \mathcal{R}_G(s_2, a)| \leq \varepsilon \text{ AND } \forall_{s_A \in \mathcal{S}_A} \left| \sum_{s_G' \in G(s_A)} [\mathcal{T}_G(s_1, a, s_G') - \mathcal{T}_G(s_2, a, s_G')] \right| \leq \varepsilon. \quad (24)$$

**Lemma 2.** When  $\mathcal{S}_A$  is created using a  $\tilde{\phi}_{model,\varepsilon}$  type:

$$\forall_{s \in \mathcal{S}_G} V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq \frac{2\varepsilon + 2\gamma\varepsilon(|\mathcal{S}_G| - 1)}{(1 - \gamma)^3}. \quad (25)$$

**Proof of Lemma 2:**

Let  $B$  be the maximum  $Q$ -value difference between any pair of ground states in the same abstract state for  $\tilde{\phi}_{model,\varepsilon}$ :

$$B = \max_{s_1, s_2, a} |Q_G(s_1, a) - Q_G(s_2, a)|,$$

where  $s_1, s_2 \in G(s_A)$ . First, we expand:

$$B = \max_{s_1, s_2, a} \left| \mathcal{R}_G(s_1, a) - \mathcal{R}_G(s_2, a) + \gamma \sum_{s_G' \in \mathcal{S}_G} \left[ (\mathcal{T}_G(s_1, a, s_G') - \mathcal{T}_G(s_2, a, s_G')) \max_{a'} Q_G(s_G', a') \right] \right| \quad (26)$$

Since difference of rewards is bounded by  $\varepsilon$ :

$$B \leq \varepsilon + \gamma \sum_{s_A \in \mathcal{S}_A} \sum_{s_G' \in G(s_A)} \left[ (\mathcal{T}_G(s_1, a, s_G') - \mathcal{T}_G(s_2, a, s_G')) \max_{a'} Q_G(s_G', a') \right]. \quad (27)$$

By similarity of transitions under  $\tilde{\phi}_{model,\varepsilon}$ :

$$B \leq \varepsilon + \gamma \text{QMAX} \sum_{s_A \in \mathcal{S}_A} \varepsilon \leq \varepsilon + \gamma |\mathcal{S}_G| \varepsilon \text{QMAX}.$$

Recall that  $\text{QMAX} = \frac{\text{RMAX}}{1 - \gamma}$ , and we defined  $\text{RMAX} = 1$ :

$$B \leq \frac{\varepsilon + \gamma(|\mathcal{S}_G| - 1)\varepsilon}{1 - \gamma}.$$

Since the  $Q$ -values of ground states grouped under  $\tilde{\phi}_{model,\varepsilon}$  are strictly less than  $B$ , we can understand  $\tilde{\phi}_{model,\varepsilon}$  as a type of  $\tilde{\phi}_{Q^*,B}$ . Applying Lemma 1 yields Lemma 2.  $\square$

### 5.4 Boltzmann over Optimal Q: $\tilde{\phi}_{bolt,\varepsilon}$

Here, we introduce  $\tilde{\phi}_{bolt,\varepsilon}$ , which aggregates states with similar Boltzmann distributions on  $Q$ -values. This type of abstractions is appealing as Boltzmann distributions balance exploration and exploitation [32]. We find this type particularly interesting for abstraction purposes as, unlike  $\tilde{\phi}_{Q^*,\varepsilon}$ , it allows for aggregation when  $Q$ -value ratios are similar but their magnitudes are different.

**Definition 14** ( $\tilde{\phi}_{bolt,\varepsilon}$ ): We let  $\tilde{\phi}_{bolt,\varepsilon}$  define a type of abstractions that, for fixed  $\varepsilon$ , satisfies:

$$\tilde{\phi}_{bolt,\varepsilon}(s_1) = \tilde{\phi}_{bolt,\varepsilon}(s_2) \rightarrow \forall a \left| \frac{e^{Q_G(s_1,a)}}{\sum_b e^{Q_G(s_1,b)}} - \frac{e^{Q_G(s_2,a)}}{\sum_b e^{Q_G(s_2,b)}} \right| \leq \varepsilon. \quad (28)$$

We also assume that the difference in normalizing terms is bounded by some non-negative constant,  $k_{bolt} \in \mathbb{R}$ , of  $\varepsilon$ :

$$\left| \sum_b e^{Q_G(s_1,b)} - \sum_b e^{Q_G(s_2,b)} \right| \leq k_{bolt} \times \varepsilon. \quad (29)$$

**Lemma 3.** When  $S_A$  is created using a function of the  $\tilde{\phi}_{bolt,\varepsilon}$  type, for some non-negative constant  $k \in \mathbb{R}$ :

$$\forall s \in S_G V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq \frac{2\varepsilon \left( \frac{|A|}{1-\gamma} + \varepsilon k_{bolt} + k_{bolt} \right)}{(1-\gamma)^2}. \quad (30)$$

We use the approximation for  $e^x$ , with  $\delta$  error:

$$e^x = 1 + x + \delta \approx 1 + x. \quad (31)$$

We let  $\delta_1$  denote the error in approximating  $e^{Q_G(s_1,a)}$  and  $\delta_2$  denote the error in approximating  $e^{Q_G(s_2,a)}$ .

**Proof of Lemma 3:**

By the approximation in Equation 31 and the assumption in Equation 29:

$$\left| \frac{1 + Q_G(s_1,a) + \delta_1}{\sum_j e^{Q_G(s_1,a_j)}} - \frac{1 + Q_G(s_2,a) + \delta_2}{\sum_j e^{Q_G(s_1,a_j)} \underbrace{\pm k\varepsilon}_{\textcircled{a}}} \right| \leq \varepsilon \quad (32)$$

Either term  $\textcircled{a}$  is positive or negative. First suppose the former. It follows by algebra that:

$$-\varepsilon \leq \frac{1 + Q_G(s_1,a) + \delta_1}{\sum_j e^{Q_G(s_1,a_j)}} - \frac{1 + Q_G(s_2,a) + \delta_2}{\sum_j e^{Q_G(s_1,a_j)} + \varepsilon k_{bolt}} \leq \varepsilon \quad (33)$$

Moving terms:

$$\begin{aligned} -\varepsilon \left( k\varepsilon + \sum_j e^{Q_G(s_1,a_j)} \right) - \delta_1 + \delta_2 \leq \\ \varepsilon k_{bolt} \left( \frac{1 + Q_G(s_1,a) + \delta_1}{\sum_j e^{Q_G(s_1,a_j)}} \right) + Q_G(s_1,a) - Q_G(s_2,a) \leq \\ \varepsilon \left( \varepsilon k_{bolt} + \sum_j e^{Q_G(s_1,a_j)} \right) - \delta_1 + \delta_2 \end{aligned} \quad (34)$$

When  $\textcircled{a}$  is the negative case, it follows that:

$$-\varepsilon \leq \frac{1 + Q_G(s_1,a) + \delta_1}{\sum_j e^{Q_G(s_1,a_j)}} - \frac{1 + Q_G(s_2,a) + \delta_2}{\sum_j e^{Q_G(s_1,a_j)} - \varepsilon k_{bolt}} \leq \varepsilon \quad (35)$$

By similar algebra that yielded Equation 34:

$$\begin{aligned}
-\varepsilon \left( -\varepsilon k_{\text{bolt}} + \sum_j e^{Q_G(s_1, a_j)} \right) - \delta_1 + \delta_2 \leq \\
-k\varepsilon \left( \frac{1 + Q_G(s_1, a) + \delta_1}{\sum_j e^{Q_G(s_1, a_j)}} \right) + Q_G(s_1, a) - Q_G(s_2, a) \leq \\
\varepsilon \left( \varepsilon k_{\text{bolt}} + \sum_j e^{Q_G(s_1, a_j)} \right) - \delta_1 + \delta_2 \quad (36)
\end{aligned}$$

Combining Equation 34 and Equation 36 results in:

$$|Q_G(s_1, a) - Q_G(s_2, a)| \leq \varepsilon \left( \frac{|\mathcal{A}|}{1-\gamma} + \varepsilon k_{\text{bolt}} + k_{\text{bolt}} \right). \quad (37)$$

Consequently, we can consider  $\tilde{\phi}_{\text{bolt}, \varepsilon}$  as a special case of the  $\tilde{\phi}_{Q^*, B}$  type, where  $B = \left( \frac{|\mathcal{A}|}{1-\gamma} + \varepsilon k_{\text{bolt}} + k_{\text{bolt}} \right)$ . Lemma 3 then follows from Lemma 1.  $\square$

## 5.5 Multinomial over Optimal Q: $\tilde{\phi}_{\text{mult}, \varepsilon}$

We consider approximate abstractions derived from a multinomial distribution over  $Q^*$  for similar reasons to the Boltzmann distribution. Additionally, the multinomial distribution is appealing for its simplicity.

**Definition 15** ( $\tilde{\phi}_{\text{mult}, \varepsilon}$ ): We let  $\tilde{\phi}_{\text{mult}, \varepsilon}$  define a type of abstraction that, for fixed  $\varepsilon$ , satisfies

$$\tilde{\phi}_{\text{mult}, \varepsilon}(s_1) = \tilde{\phi}_{\text{mult}, \varepsilon}(s_2) \rightarrow \forall_a \left| \frac{Q_G(s_1, a)}{\sum_b Q_G(s_1, b)} - \frac{Q_G(s_2, a)}{\sum_b Q_G(s_2, b)} \right| \leq \varepsilon. \quad (38)$$

We also assume that the difference in normalizing terms is bounded by some non-negative constant,  $k_{\text{mult}} \in \mathbb{R}$ , of  $\varepsilon$ :

$$\left| \sum_i Q_G(s_1, a_i) - \sum_j Q_G(s_2, a_j) \right| \leq k_{\text{mult}} \times \varepsilon. \quad (39)$$

**Lemma 4.** When  $S_A$  is created using a function of the  $\tilde{\phi}_{\text{mult}, \varepsilon}$  type, for some non-negative constant  $k_{\text{mult}} \in \mathbb{R}$ :

$$\forall_{s \in S_M} V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq \frac{2\varepsilon \left( \frac{|\mathcal{A}|}{1-\gamma} + k_{\text{mult}} \right)}{(1-\gamma)^2} \quad (40)$$

**Proof of Lemma 4**

The proof follows an identical strategy to that of Lemma 3, but without the approximation  $e^x \approx 1 + x$ .  $\square$

## 6 Example Domains

We apply approximate abstraction to five example domains—NChain, Upworld, Taxi, Minefield and Random. These domains were selected for their diversity—NChain is relatively simple, Upworld is particularly illustrative of the power of abstraction, Taxi is goal-based and hierarchical in nature, Minefield is stochastic, and Random MDP has many near-optimal policies.

Our code base<sup>1</sup> provides implementations for abstracting arbitrary MDPs as well as visualizing and evaluating the resulting abstract MDPs. We use the graph-visualization library GraphStream [29] and the planning and RL library, BURLAP<sup>2</sup>. For all experiments, we set  $\gamma$  to 0.95.

<sup>1</sup>[https://github.com/david-abel/state\\_abstraction](https://github.com/david-abel/state_abstraction)

<sup>2</sup><http://burlap.cs.brown.edu/>

## 6.1 Visualizations

We provide visuals of both the ground MDP and resulting abstract MDP for each domain. A grey circle indicates a state and colored arrows indicate transitions. The thickness of the arrow indicates how much reward is associated with that transition. In the ground MDPs, states are labeled with a number. In the abstract MDPs, we indicate which ground states were collapsed to each abstract state by labelling the abstract states with their ground states.

### 6.1.1 NChain

NChain is a simple MDP investigated in the Bayesian RL literature due to the interesting exploration problem it poses [11]. In our implementation, we set  $N = 10$ , normalized rewards between 0 and 1, and used a slip probability of 0.2. An NChain instance ( $N = 10$ ) and its abstraction are visualized in Figure 6.1.2.

In all states, the agent has two actions available: advance down the chain, or return to state 0. The agent receives .2 reward for returning to state 0, and no reward for advancing down the chain. The exception is that when the agent transitions to the last state in the chain, it receives 1.0 reward. Transitions also have small slip probability  $\rho$ , such that the applied action results in the opposite dynamics. In our implementation, we set  $N = 10$  and  $\rho = 0.2$ .

### 6.1.2 Upworld

The Upworld task is an  $N \times M$  grid in which the agent starts in the lower left corner. The agent may move left, right, and up. The agent receives positive reward for transitioning to any state at the top of the grid, where moving up in the top cells self transitions. the agent receives 0 reward for all other transitions. Consequently, moving up is always the optimal action, since moving left and right does not change the agent’s manhattan distance to positive reward. During experimentation, we set  $N = 10$ ,  $M = 4$ . An Upworld instance ( $N = 10$ ,  $M = 4$ ) and its abstraction are visualized in Figure 6.1.2.

Upworld illustrates a compelling property with respect to state abstraction: the optimal *exact*  $Q^*$  abstraction function (when  $\varepsilon = 0$ ) can always construct an abstract MDP with  $|\mathcal{S}_A| = N$ , the height of the grid, with no change in the value of the optimal policy. Consequently, letting  $M$  be arbitrarily large, Upworld offers an arbitrary reduction in the size of the MDP through abstraction, at no cost to the value of the optimal policy. This is a result of the property that all states in the same row have the same  $Q$  values:

**Remark:** *The optimal exact abstraction,  $\phi_{Q^*,0}$ , induces an abstract MDP with an optimal policy of equal value to the true optimal policy, and reduces the size of the state space from  $N \times M$  (ground) to  $N$  (abstract).*

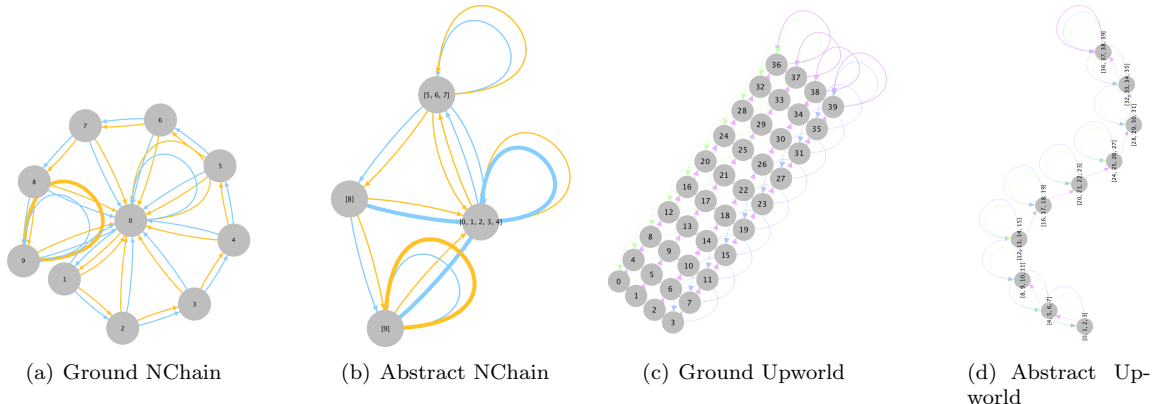


Figure 2: Comparison of the ground and abstract MDPs, under  $\tilde{\phi}_{Q^*,\varepsilon}$ , with  $\varepsilon = 0.5$

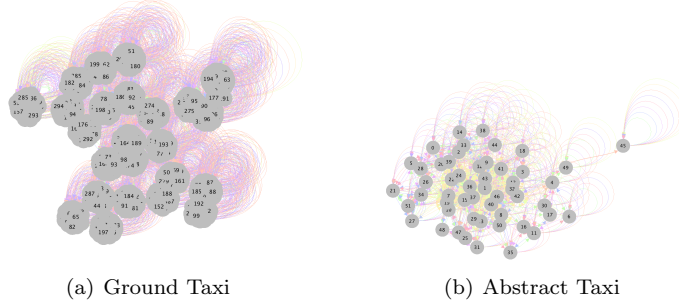


Figure 3: Comparison of the ground and abstract Taxi MDPs under an  $\tilde{\phi}_{Q^*,\varepsilon}$  abstraction, with  $\varepsilon = 0.03$ .

### 6.1.3 Taxi

Taxi has long been studied by the hierarchical RL literature [12]. The agent, operating in a Grid World style domain [30], may move left, right, up, and down, as well as pick up a passenger and drop off a passenger. The goal is achieved when the agent has taken all passengers to their destinations.

We visualize the compression on a simple 626 Taxi instance in Figure 3. As stated above, we visualize the original Taxi problem into a graph representation so that we may visualize both the ground MDP and abstract MDP in the same format, despite the unnatural appearance.

### 6.1.4 Minefield

Minefield is a test problem we are introducing that uses the Grid World dynamics of Russell and Norvig [30] with slip probability of  $x$ . The reward function is such that moving up in the top row of the grid receives 1.0 reward; all other transitions receive 0.2 reward, except for transitions to a random set of  $\kappa$  mine-states (which may include the top row) that receive 0 reward. We set  $N = 10, M = 4, \varepsilon = 0.5, \kappa = 5, x = 0.01$ .

### 6.1.5 Random MDP

In the Random MDP domain we consider, there are 100 states and 3 actions. For each state, each action transitions to one of two randomly selected states with probability 0.5. The Random MDP and its compression are visualized in Figure 4.

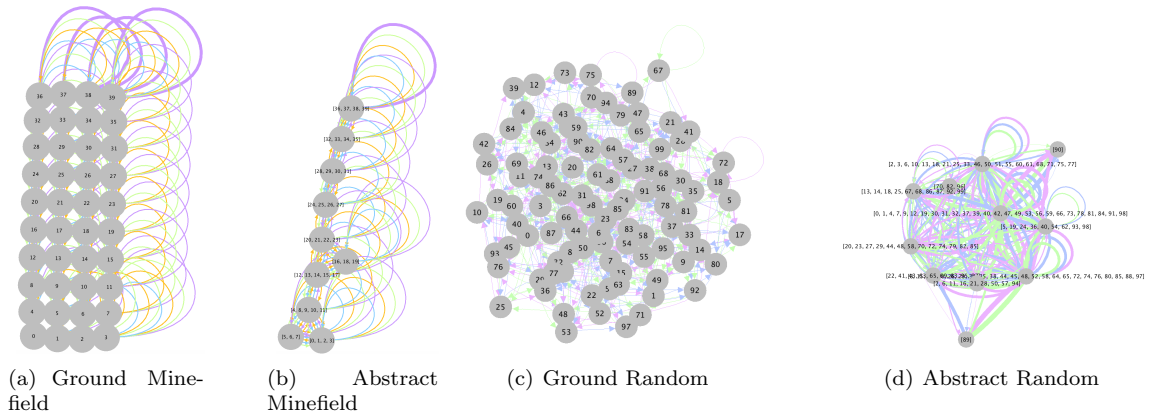


Figure 4: Comparison of the ground and abstract MDPs under an  $\tilde{\phi}_{Q^*,\varepsilon}$  abstraction, with  $\varepsilon = 0.5$ .

## 7 Empirical Results

We ran experiments on the  $\tilde{\phi}_{Q^*,\varepsilon}$  type aggregation functions. We provide results for only  $\tilde{\phi}_{Q^*,\varepsilon}$  because, as our proofs in Section 5 demonstrate, the other three functions are reducible to particular  $\tilde{\phi}_{Q^*,\varepsilon}$  functions. For the purpose of illustrating what kinds of approximations are possible we built each abstraction by first solving the MDP, then greedily aggregating ground states into abstract states that satisfied the  $\tilde{\phi}_{Q^*,\varepsilon}$  criteria. Since this approach represents an order-dependent approximation to the maximum amount of abstraction possible, we randomized the order in which states were considered across trials. Every ground state is equally weighted in its abstract state.

For each domain, we report two quantities as a function of epsilon with 95% confidence bars. First, we compare the number of states in the abstract MDP for different values of  $\varepsilon$ , shown in the left column of Figure 5 and Figure 6. The smaller the number of abstract states, the smaller the state space of the MDP that the agent must plan over. Second, we report the value under the abstract policy of the initial ground state, also shown in the right column of Figure 5 and Figure 6. In the Taxi and Random domains, 200 trials were run for each data point, whereas 20 trials were sufficient in Upworld, Minefield, and NChain.

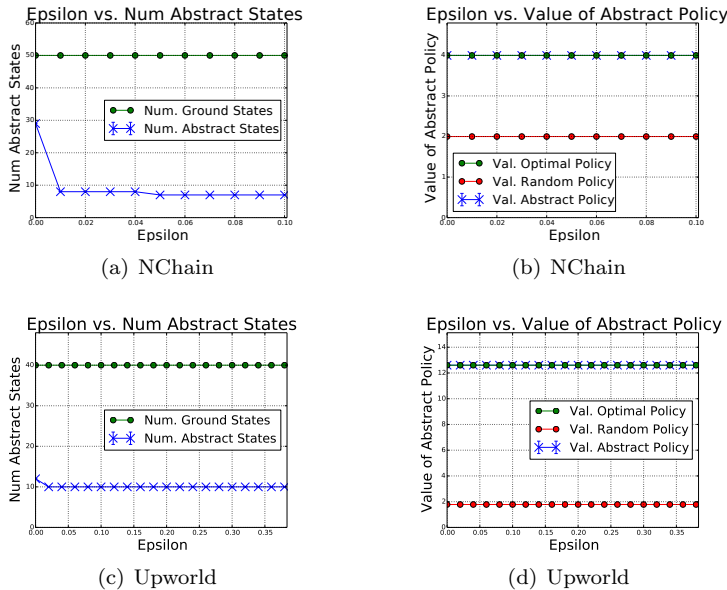


Figure 5:  $\varepsilon$  vs. Num States (left) and  $\varepsilon$  vs. Abstract Policy Value (right).

Our empirical results corroborate our thesis—approximate state abstractions can decrease state space size while retaining bounded error. In both NChain and Minefield, we observe that, as  $\varepsilon$  increases from 0, the number of states that must be planned over is reduced, and optimal behavior is either fully maintained (NChain) or very nearly maintained (Minefield). Similarly for Taxi, when  $\varepsilon$  is between .02 and .025, we observe a reduction in the number of states in the abstract MDP while value is fully maintained. After .025, increased reduction in state space size comes at a cost of value. Lastly, as  $\varepsilon$  is increased in the Random domain, there is a smooth reduction in the number of abstract states with a corresponding cost in the value of the derived policy. When  $\varepsilon = 0$ , there is no reduction in state space size whatsoever (the ground MDP has 100 states), because no two states have identical optimal  $Q$ -values.

Our experimental results also highlight a noteworthy characteristic of approximate state abstraction in goal-based MDPs. Taxi exhibits relative stability in state space size and behavior for  $\varepsilon$  up to .02, at which point both fall off dramatically. We attribute the sudden fall off of these quantities to the goal-based nature of the domain; once information critical for achieving optimal behavior is lost in the state aggregation,

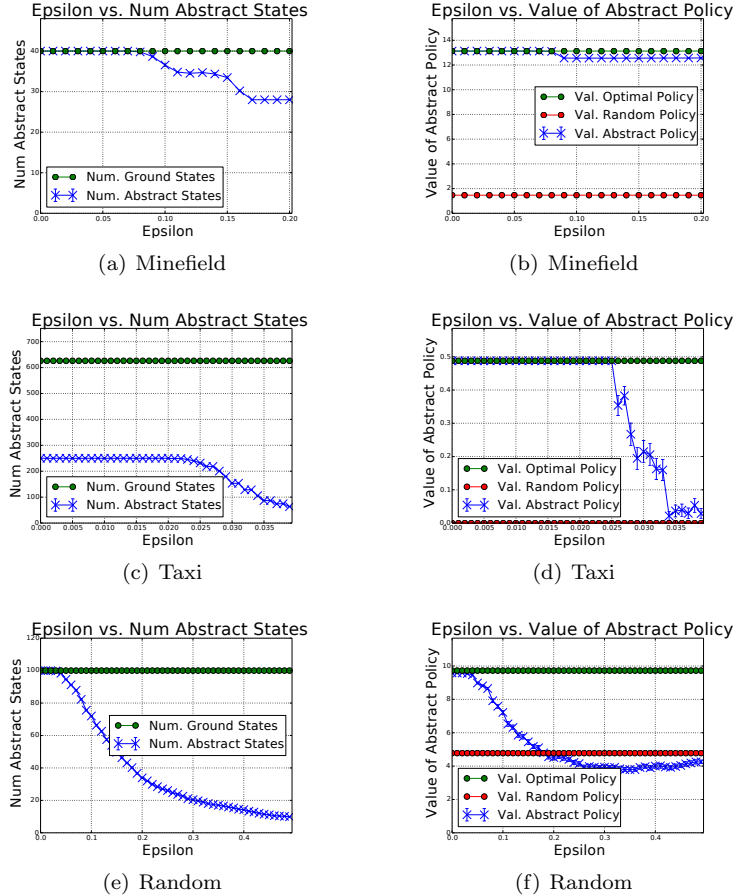


Figure 6:  $\epsilon$  vs. Num States (left) and  $\epsilon$  vs. Abstract Policy Value (right).

solving the goal—and so acquiring any reward—is impossible. Conversely, in the Random domain, a great deal of near optimal policies are available to the agent. Thus, even as the information for optimal behavior is lost, there are many near optimal policies available to the agent that remain available.

## 8 Conclusion

Approximate abstraction in MDPs offers considerable advantages over exact abstraction. First, approximate abstraction relies on criteria that we imagine a planning or learning algorithm to be able to learn without solving the full MDP. Second, approximate abstractions can achieve greater degrees of compression due to their relaxed criteria of equality. Third, methods that employ approximate aggregation techniques are able to tune the aggressiveness of abstraction all the while incurring bounded error. In this work, we proved bounds for the value lost when behaving according to the optimal policy of the abstract MDP, and empirically demonstrate that approximate abstractions can reduce state space size with minor loss in the quality of the behavior. We provide a code base that provides implementations to abstract, visualize, and evaluate an arbitrary MDP to promote further investigation into approximate abstraction.

There are many directions for future work. First, we are interested in extending the approach of Ortner [27] by learning the approximate abstraction functions introduced in this paper online in the planning or RL setting, particularly when the agent must solve a collection of related MDPs. Additionally, while



our work presents several sufficient conditions for achieving bounded error of learned behavior with approximate abstractions, we hope to investigate what conditions are strictly necessary for an approximate abstraction to achieve bounded error. Further, we are interested in characterizing the relationship between temporal abstractions, such as options [33] and approximate state abstractions. Lastly, we are interested in understanding the relationship between various approximate abstractions and the information theoretical limitations on the degree of abstraction achievable in MDPs.

## References

- [1] David Abel, David Ellis Hershkowitz, Gabriel Barth-Maron, Stephen Brawner, Kevin O’Farrell, James MacGlashan, and Stefanie Tellex. Goal-based action priors. In *ICAPS*, pages 306–314, 2015.
- [2] David Andre and Stuart J Russell. State abstraction for programmable reinforcement learning agents. In *AAAI/IAAI*, pages 119–125, 2002.
- [3] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 89–96, 2009.
- [4] Bram Bakker and Jürgen Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, pages 438–445, 2004.
- [5] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003.
- [6] James C Bean, John R Birge, and Robert L Smith. Dynamic programming aggregation. *Operations Research*, 35(2):215–220, 2011.
- [7] Peter Dayan and Geoffrey Hinton. Feudal Reinforcement Learning. *Advances in neural information processing systems*, pages 271–278, 1993.
- [8] Thomas Dean and Robert Givan. Model minimization in markov decision processes. In *AAAI/IAAI*, pages 106–111, 1997.
- [9] Thomas Dean, Robert Givan, and Sonia Leach. Model reduction techniques for computing approximately optimal solutions for markov decision processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 124–131. Morgan Kaufmann Publishers Inc., 1997.
- [10] Richard Dearden and Craig Boutilier. Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, 89(1):219–283, 1997.
- [11] Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998.
- [12] Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.
- [13] Eyal Even-Dar and Yishay Mansour. Approximate equivalence of Markov decision processes. In *Learning Theory and Kernel Machines*, pages 581–594. Springer, 2003.
- [14] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 162–169. AUAI Press, 2004.
- [15] Norman Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. Methods for computing state similarity in markov decision processes. *Proceedings of the 22nd conference on Uncertainty in artificial intelligence*, 2006.

- [16] Jesse Hostetler, Alan Fern, and Tom Dietterich. State Aggregation in Monte Carlo Tree Search. *Aaai 2014*, page 7, 2014.
- [17] Marcus Hutter. Extreme state aggregation beyond mdps. In *International Conference on Algorithmic Learning Theory*, pages 185–199. Springer, 2014.
- [18] Marcus Hutter. Extreme state aggregation beyond markov decision processes. *Theoretical Computer Science*, 650:73–91, 2016.
- [19] Nan Jiang. Abstraction Selection in Model-Based Reinforcement Learning. *icml*, 37, 2015.
- [20] Nan Jiang, Satinder Singh, and Richard Lewis. Improving uct planning via approximate homomorphisms. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1289–1296. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [21] Nicholas K Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *IJCAI*, pages 752–757, 2005.
- [22] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, pages 237–285, 1996.
- [23] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.
- [24] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *ISAAC*, 2006.
- [25] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.
- [26] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popovic. Efficient bayesian clustering for reinforcement learning. *IJCAI*, 2016.
- [27] Ronald Ortner. Adaptive aggregation for reinforcement learning in average reward Markov decision processes. *Annals of Operations Research*, 208(1):321–336, 2013.
- [28] Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.
- [29] Yoann Pigné, Antoine Dutot, Frédéric Guinand, and Damien Olivier. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. *CoRR*, abs/0803.2093, 2008.
- [30] Stuart Russell and Peter Norvig. *Artificial Intelligence A Modern Approach*. Prentice-Hall, Englewood Cliffs, 1995.
- [31] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement Learning in Finite MDPs : PAC Analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.
- [32] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [33] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.
- [34] Thomas J Walsh, Lihong Li, and Michael L Littman. Transferring state abstractions between mdps. In *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.