State Abstraction as Compression in Apprenticeship Learning

David Abel,[1] **Dilip Arumugam,**[2] **Kavosh Asadi,**[1]
Yuu Jinnai,[1] **Michael L. Littman,**[1] **Lawson L.S. Wong**[3]
[1]Department of Computer Science, Brown University
[2]Department of Computer Science, Stanford University
[3]College of Computer and Information Science, Northeastern University

Abstract

State abstraction can give rise to models of environments that are both compressed and useful, thereby enabling efficient sequential decision making. In this work, we offer the first formalism and analysis of the trade-off between compression and performance made in the context of state abstraction for Apprenticeship Learning. We build on Rate-Distortion theory, the classic Blahut-Arimoto algorithm, and the Information Bottleneck method to develop an algorithm for computing state abstractions that approximate the optimal trade-off between compression and performance. We illustrate the power of this algorithmic structure to offer insights into effective abstraction, compression, and reinforcement learning through a mixture of analysis, visuals, and experimentation.

1 Introduction

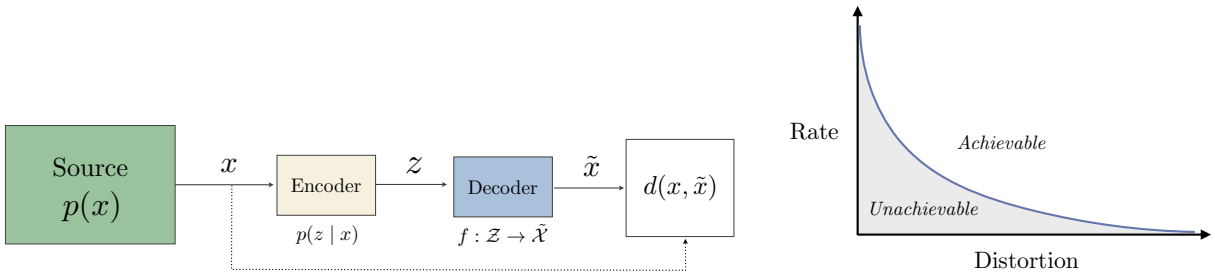
Reinforcement Learning (RL) poses a challenging problem. Agents must learn about their environment through high-dimensional and often noisy observations while receiving sparse and delayed evaluative feedback. The ability to understand one's surroundings well enough to support effective decision making under these conditions is a remarkable feat, and arguably a hallmark of intelligent behavior. To this end, a long-standing goal of RL is to endow decision-making agents with the ability to acquire and exploit abstract models for use in decision making, drawing inspiration from human cognition (Tenenbaum et al. 2011).

One path toward realizing this goal is to make use of *state abstraction*, which describes methods for compressing the environment's state space to distill complex problems into simpler forms (Dietterich 2000b; Andre and Russell 2002; Li, Walsh, and Littman 2006). Critically, the degree of compression induced by an abstraction trades off directly with its capacity to represent good behavior. If the abstraction throws away too much information, the resulting abstract model will fail to preserve essential characteristics of the original task (Abel, Hershkowitz, and Littman 2016). Thus, care must be taken to identify state abstractions that balance between an appropriate degree of compression and adequate representational power.

Information Theory offers foundational results about the limits of compression (Shannon 1948). The core of the theory clarifies how to communicate in the presence of noise, culminating in seminal results about the nature of communication and compression that helped establish the science and engineering practices of computation. Of particular relevance to our agenda is Rate-Distortion theory, which studies the trade-off between a code's ability to compress (rate) and represent the original signal (distortion) (Shannon 1948; Berger 1971). Cognitive neuroscience has suggested that perception and generalization are tied to efficient compression (Attneave 1954; Sims 2016; 2018), termed the "efficient coding hypothesis" by Barlow (1961).

The goal of this work is to understand the role of information-theoretic compression in state abstraction for sequential decision making. We draw a parallel between *state abstraction* as used in reinforcement learning and *compression* as understood in information theory. We build on the seminal work of Shannon (1948), Blahut (1972), Arimoto (1972) and Tishby, Pereira, and Bialek (1999), and draw inspiration from related work on understanding the relationship between abstraction and compression (Botvinick et al. 2015; Solway et al. 2014). While the perspective we introduce is intended to be general, we focus our study in two ways. First, we investigate only *state abstraction*, deferring discussion of temporal (Sutton, Precup, and Singh 1999), action (Hauskrecht et al. 1998), and hierarchical abstraction (Dayan and Hinton 1993; Dietterich 2000a) to future work. Second, we address the learning problem when a *demonstrator* is available, as in Apprenticeship Learning (Atkeson and Schaal 1997; Abbeel and Ng 2004; Argall et al. 2009), which simplifies aspects of our model.

Concretely, we introduce a new objective function that explicitly balances state-compression and performance. Our main result proves this objective is upper bounded by a variant of the Information Bottleneck objective adapted to sequential decision making. We introduce Deterministic Information Bottleneck for State abstraction (DIBS), an algorithm that outputs a *lossy* state abstraction optimizing the trade-off between compressing the state space and preserving the capacity for performance in that compressed state space. We conduct experiments to showcase the relationship between compression and performance captured by the algorithm in a traditional grid world and present an extension to high-dimensional observations via experiments with the Atari game Breakout.



(a) RD Setting

(b) RD Lower Bound

Figure 1: The usual Rate-Distortion setting (left) and the lower bound on the Rate-Distortion trade-off (right)

2 Background

Reinforcement Learning (RL) is the problem of an agent learning to make decisions in an environment through interaction alone. We assume the standard formalism: an agent interacts with a Markov Decision Process (MDP) to maximize long-term expected discounted reward. Additionally, we focus on Apprenticeship Learning, wherein an agent observes the policy π_E of an expert who is trying to maximize an unknown reward function. For more on MDPs, see (Puterman 2014), for RL, see (Sutton and Barto 2018), and for Apprenticeship Learning, see (Atkeson and Schaal 1997; Abbeel and Ng 2004; Argall et al. 2009).

Abstraction serves as a powerful tool for lowering the complexity of decision making. Abstraction appears in several forms: state, action/temporal, and hierarchical. In all cases, abstraction is used to distill the agent's representation of the environment to information that is most useful for making effective decisions. Naturally, state abstraction is tightly connected to compression. State abstraction encompasses methods that aggregate states together to form abstract states (Whitt 1978; Bertsekas and Castanon 1989; Singh, Jaakkola, and Jordan 1995; McCallum 1996; Dean, Givan, and Leach 1997; Dietterich 2000b; Andre and Russell 2002; Givan, Dean, and Greig 2003; Ferns, Panangaden, and Precup 2004; Li, Walsh, and Littman 2006; Jiang, Singh, and Lewis 2014; Hostetler, Fern, and Dietterich 2014; Abel, Hershkowitz, and Littman 2016; Abel et al. 2018; Taïga, Courville, and Bellemare 2018). Given an MDP with state space $\mathcal{S}$, a state abstraction is a function ϕ that projects ground states $s \in \mathcal{S}$ to abstract states $s_\phi \in \mathcal{S}_\phi$, where typically $|\mathcal{S}_\phi| \ll |\mathcal{S}|$. In this work, we follow the formalisms of Li, Walsh, and Littman (2006) and Abel et al. (2018).

Information Theory studies communication in the presence of noise (Shannon 1948). In Shannon's words: "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." Information Theory typically investigates coder-decoder pairs and their capacity to faithfully communicate messages with zero or low error, even in the presence of noise.

Rate-Distortion (RD) Theory studies the trade-off between a coder-decoder pair's ability to compress a signal and the pair's ability to faithfully reproduce the original signal. The typical RD setting is pictured in Figure 1a: an information source generates $x \in \mathcal{X}$, which is coded via $p(z \mid x)$ for $z \in \mathcal{Z}$, and decoded via a deterministic function $f : \mathcal{Z} \to \tilde{\mathcal{X}}$. *Distortion* is defined with respect to some chosen distortion metric, $d : \mathcal{X} \times \tilde{\mathcal{X}} \to \mathbb{R}_{\geq 0}$, where typically $\mathcal{X} = \tilde{\mathcal{X}}$. The information *rate*, R, denotes the number of bits in each code word. So, with a coding alphabet $\tilde{\mathcal{Z}} = \{0, 1\}^n$, the rate is n. Shannon and Kolmogorov (see Berger (1971) for more background) offer a lower bound on the trade-off between Rate and Distortion: given a level of distortion, D, the following function defines the smallest rate that achieves expected distortion of at most D:

$$R(D) = \min_{p(\tilde{x}|x):\mathbb{E}[d(x,\tilde{x})]\leq D} I(X; \tilde{X}), \tag{1}$$

where $I(X; \tilde{X})$ is the mutual information between random variables X and $\tilde{X}$:

$$I(X; \tilde{X}) := \sum_{x \in \mathcal{X}} \sum_{\tilde{x} \in \tilde{\mathcal{X}}} p(x, \tilde{x}) \log \frac{p(x, \tilde{x})}{p(x)p(\tilde{x})}. \tag{2}$$

Intuitively, Equation 1 tells us that as we add bits to our code, we can more faithfully reconstruct our source messages. The curve displayed in Figure 1b shows an example lower bound of the trade-off between Rate and Distortion expressed by Equation 1.

For a given information source, it is natural to consider how one might compute the coder-decoder pair that achieves one of the minimal points defined by the Rate-Distortion function. Finding this point presents the following optimization problem:

$$\min_{p(\tilde{x}|x)} \underbrace{I(X; \tilde{X})}_{Rate} + \beta \underbrace{\mathbb{E}_{p(x,\tilde{x})}\left[d(x, \tilde{x})\right]}_{Distortion}, \tag{3}$$

with a Lagrange multiplier $\beta \in \mathbb{R}_{\geq 0}$ expressing the relative preference between preserving rate and distortion. As β gets closer to 0, rate becomes more important, while as β approaches ∞, minimizing distortion is prioritized. Blahut-Arimoto (BA) is a simple iterative algorithm that converges to the global optimum of this optimization problem (Arimoto 1972; Blahut 1972). BA alternates between the fol-

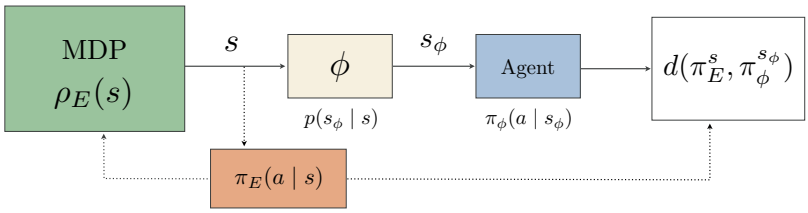


Figure 2: Our framework for trading off compression with value via state abstraction.

lowing two steps, for a given $\beta \in \mathbb{R}_{\geq 0}$:

$$p_{t+1}(\tilde{x}) = \sum_{x \in \mathcal{X}} p(x) p_t(\tilde{x} \mid x)$$

$$p_{t+1}(\tilde{x} \mid x) = \frac{p_{t+1}(\tilde{x}) \exp\{-\beta d(x, \tilde{x})\}}{\sum_{x' \in \tilde{\mathcal{X}}} p_{t+1}(x') \exp\{-\beta d(x, x')\}}.$$

BA is known to converge to the global optimum with convergence rate:

$$O\left(|\mathcal{X}||\tilde{X}|\sqrt{\log(|\tilde{X}|)}/\varepsilon\right), \tag{4}$$

for ε error tolerance (Arimoto 1972). The computational complexity of finding the exact solution for a discrete, memoryless channel is unknown. For a continuous memoryless channel, the problem is an infinite-dimensional convex optimization which is known to be NP-Hard (Sutter et al. 2015).

The Information Bottleneck (IB) Method extends RD theory to prediction. Traditional RD theory defines "relevant" information by choice of a distortion function—codes are said to capture relevant information if they achieve low distortion. The IB defines relevant information according to how well a random variable Y can be predicted from each $\tilde{x} \in \tilde{\mathcal{X}}$. For the IB to make sense, we must suppose that $I(X;Y) > 0$, and that the coder–decoder scheme has access to the joint probability mass function (pmf) $p(x, y)$. IB then recasts the RD lower bound in Equation 1 in terms of prediction of Y given $\tilde{X}$. The optimal assignment to the distribution $p(\tilde{x} \mid x)$ is then given by minimizing:

$$\mathcal{L}\left[p(\tilde{x} \mid x)\right] = I(\tilde{X}; X) - \beta I(\tilde{X}; Y), \tag{5}$$

where $\beta \in \mathbb{R}_{\geq 0}$ is again a Lagrange multiplier attached to the meaningful information. Like BA, choice of β determines the relative preference between compression (rate) and predicting Y (distortion); when $\beta = 0$, the coder can ignore Y entirely, and so is free to compress arbitrarily. Conversely, as $\beta \to \infty$, the coder must prioritize prediction of Y, requiring more bits in the coding alphabet.

Tishby, Pereira, and Bialek (1999) offer a convergent algorithm for solving the above optimization problem.

Theorem 1 *(Appears as Theorem 5 in Tishby, Pereira, and Bialek (1999)) Equation 5 yields the following optimization problem:*

$$\min_{p(\tilde{x}|x)} \mathcal{L}_{IB}\left[p(\tilde{x} \mid x); p(\tilde{x}); p(y \mid \tilde{x})\right] = \tag{6}$$

$$\min_{p(\tilde{x}|x)} \left(I(X; \tilde{X}) + \beta \mathop{\mathbb{E}}_{p(x,\tilde{x})}\left[D_{KL}(p(y \mid x) \,||\, p(y \mid \tilde{x})\right]\right),$$

where D_{KL} denotes the Kullback-Leibler (KL) Divergence:

$$D_{KL}(p \,||\, q) := \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}, \tag{7}$$

for p and q pmfs with overlapping support $\mathcal{X}$.

The algorithm consists of the following three steps, which, when repeated, converge to a local minima of the above optimization problem, with $Z(\beta, x)$ a normalizing term:

$$\begin{cases} p_t(\tilde{x} \mid x) \leftarrow \frac{p_t(\tilde{x})}{Z(\beta,x)} \exp\{-\beta D_{KL}(p(y \mid x) \,||\, p_t(y \mid \tilde{x}))\}, \\ p_{t+1}(\tilde{x}) \leftarrow \sum_x p(x) p_t(\tilde{x} \mid x), \\ p_{t+1}(y \mid \tilde{x}) \leftarrow \sum_y p(y \mid x) p_t(x \mid \tilde{x}). \end{cases}$$

It is important to note that the algorithm only presents a *locally optimal* solution to the above optimization problem. To the best of our knowledge, there is no known efficient algorithm for computing the global optimum. Mumey and Gedeon (2003) show that a closely related problem to finding the global optimum in the above is in fact NP-Hard, suggesting that local convergence or approximation is likely our best option. Additionally, if the support of $p(y \mid x)$ and $p(y \mid \tilde{x})$ does not *exactly* overlap, then D_{KL} is trivially infinity, leading to vacuous updates. It is thus important that application of their algorithm be applied in a context with overlapping supports.

The Deterministic IB (DIB) extends the IB by focusing on deterministic coding functions where $p(\tilde{x} \mid x) \equiv f : \mathcal{X} \to \tilde{\mathcal{X}}$ (Strouse and Schwab 2017). Given the equality $I(X;Y) = H(X) - H(X \mid Y)$, note that when the coder is a deterministic function f, we can replace the mutual information term in the objective by the entropy of the latent space:

$$\min_{f(x)} \mathcal{L}_{\text{DIB}}\left[f(x); p(\tilde{x}); p(y \mid x)\right] =$$

$$\min_{f(x)} \left(H(\tilde{X}) + \beta \mathop{\mathbb{E}}_{p(x)}\left[D_{\text{KL}}(p(y \mid x) \,||\, p(y \mid \tilde{x}))\right]\right). \tag{8}$$

Given that state abstractions are often deterministic, we will primarily be focused on this extension.

3 State Abstraction as Compression

We now adapt the Information Bottleneck to sequential decision making. Our model for state compression is pictured in Figure 2, with the information generating source defined as ρ_E, the stationary distribution on the given MDP induced by the expert policy, π_E. That is, the source distribution is

defined as the stationary distribution $\rho_E(s)$, for each $s \in \mathcal{S}$, for a given start state $s_0 \in \mathcal{S}$, as:

$$\rho_E(s) := \sum_{t=0}^{\infty} \gamma^t \Pr\{s_t = s \mid s_0, \pi_E\} \tag{9}$$

Our goal is to answer the following question: *How many abstract states are needed for an agent to faithfully make similar decisions to an expert demonstrator?* We cast the Rate-Distortion trade-off as a trade-off between (1) the size of the abstract state space $|\mathcal{S}_\phi|$, and (2) the value of the best policy representable using $\mathcal{S}_\phi$ compared to π_E. More formally, we introduce and study the following objective:

Definition 1 (Compression-Value Abstraction or CVA Objective): *The objective function, $\mathcal{J}$, for a given Lagrange multiplier $\beta \in \mathbb{R}_{\geq 0}$, is defined as:*

$$\mathcal{J}[\phi] := |\mathcal{S}_\phi| + \beta \mathop{\mathbb{E}}_{\rho_E(s)} \left[V^{\pi_E}(s) - V^{\pi_\phi}(\phi(s)) \right]. \tag{10}$$

Our goal is to define an algorithm that minimizes the above objective in finite time. However, to the best of our knowledge, there is no known efficient method to solve the above objective. Instead, to achieve our goal, we next introduce an IB-like objective that serves as an upper bound on $\mathcal{J}$.

We use the following definition, denoting the size of the non-negligibly used portion of an alphabet under a pmf:

Definition 2 (pmf-Used Alphabet Size): *The pmf-used alphabet size of $\mathcal{X}$ is the number of elements whose probability under $p(x)$ is greater than some negligibility threshold $\delta_{min} \in (0, 1)$:*

$$|\mathcal{X}|_{p(x)}^{\delta_{min}} := \min\left\{ |\{x \in \mathcal{X} : p(x) > \delta_{min}\}|, |\mathcal{X}| \right\}.$$

This notion of alphabet size generalizes the usual method of measuring the size of a state space. When we think about the CVA objective, the state space size will typically be thought of in relation to this notion of state space size, under a given state distribution.

One might wonder why such a question cannot be answered by assigning one abstract state to each action, as is captured by the π^*-irrelevance abstractions studied by Li, Walsh, and Littman (2006). First, if the demonstrator policy is stochastic, no such abstraction exists. Second, we are ultimately interested in state abstractions that facilitate effective *learning*; if the abstraction were given to an arbitrary RL algorithm, we would like learning to be made easier. Highly aggressive abstraction types like π^* destroy guarantees and make aspects of learning harder (Li, Walsh, and Littman 2006; Abel et al. 2017). Lastly, π^*-irrelevance only captures lossless abstraction; through RD, we can build a toward theory of lossy compression for RL.

3.1 DIB Upper Bounds the CVA Objective

In our setup, the given MDP paired with the fixed control policy π_E define an information-generating source. At each time step, a state is sampled from ρ_E and given to a learning agent through a probabilistic state-abstraction function, $\phi : \mathcal{S} \to \Pr\{\mathcal{S}_\phi\}$, which projects each state to each abstract state s_ϕ with some probability. Our core simplifying assumption is: there exists a demonstrator policy π_E that controls the MDP. The agent's goal is to perform as well as the demonstrator using as small of a state space as possible, as reflected by $\mathcal{J}$.

We now construct the IB and DIB analogue objectives. First, we let $I(S; S_\phi)$ denote the rate, where S is a random variable indicating the probability of arriving in each state under ρ_E, and S_ϕ is a random variable indicating the probability of arriving in each abstract state under ρ_E and projecting through $\phi(s)$. Second, following the IB, we let $D_{\text{KL}}(\pi_E(a \mid s) \mid\mid \pi_\phi(a \mid s_\phi))$ denote the distortion. We further suppose there exists a fixed, deterministic mapping from $\mathcal{S}_\phi$ to $\tilde{\mathcal{S}}$, with $\tilde{\mathcal{S}} = \mathcal{S}$. Thus, the distribution $p(\tilde{x} \mid x)$ is simply $p(s_\phi \mid s)$, which we abbreviate as ϕ. Consequently, we find the following alignments between our objects of study (abstractions, policies) and those studied by IB:

$$p(\tilde{x} \mid x) \rightsquigarrow \phi, \quad p(\tilde{x}) \rightsquigarrow \rho_\phi, \quad p(y \mid \tilde{x}) \rightsquigarrow \pi_\phi, \tag{11}$$

where ρ_ϕ is the stationary distribution over abstract states induced by π_ϕ and ϕ. Thus, per Theorem 1, we construct an objective function $\hat{J}$ based on the IB:

$$\hat{\mathcal{J}}_{\text{IB}}[\phi; \rho_\phi; \pi_\phi] := I(S; S_\phi) + \mathop{\mathbb{E}}_{s \sim \rho_E, s_\phi \sim \phi(s)} [\beta D_{\text{KL}}(\pi_E(a \mid s) \mid\mid \pi_\phi(a \mid s_\phi))]. \tag{12}$$

If we choose to use the DIB instead, then we only consider deterministic state abstraction functions $\phi : \mathcal{S} \to \mathcal{S}_\phi$, and so $H(S_\phi \mid S) = 0$. Therefore, the DIB analogue objective is expressed as:

$$\hat{\mathcal{J}}_{\text{DIB}}[\phi; \rho_\phi; \pi_\phi] := H(S_\phi) + \mathop{\mathbb{E}}_{s \sim \rho_E} [\beta D_{\text{KL}}(\pi_E(a \mid s) \mid\mid \pi_\phi(a \mid \phi(s)))]. \tag{13}$$

We now present the main result of the paper, which relates $\hat{\mathcal{J}}_{\text{DIB}}$ to $\mathcal{J}$.

Theorem 2 *A function f of the DIB objective $\hat{\mathcal{J}}_{DIB}$ is an upper bound for the CVA Objective, $\mathcal{J}$, where state space size is treated as $|\mathcal{S}_\phi|_{\rho_\phi(s)}^{\delta_{min}}$:*

$$\forall_\phi : \mathcal{J}[\phi] \leq f\left(\hat{\mathcal{J}}_{DIB}[\phi]\right). \tag{14}$$

All proofs are presented in the supplementary material.

3.2 Relating D_{KL} and V, H and $|\mathcal{S}_\phi|$

To prove the theorem, we require two lemmas. The first relates the entropy of a pmf to the maximum size of the alphabet used by that pmf:

Lemma 1 *Consider a discrete random variable X, with alphabet $\mathcal{X}$ and some pmf $p(x)$. For a given threshold $\delta_{min} \in (0, 1)$, the pmf-used alphabet size of the alphabet is bounded, relative to some pmf $p(x)$:*

$$|\mathcal{X}|_{p(x)}^{\delta_{min}} \leq \frac{H(X)}{\delta_{min} \log\left(\frac{1}{\delta_{min}}\right)}. \tag{15}$$

This bound is relatively loose: we know trivially that $H(X) \leq \log_2 |\mathcal{X}|$. Thus, in the worst case, the bound can be up to $\tilde{O}(1/\delta_{min})$ times larger than the true alphabet. Still, this result allows us to relate the entropy of a random variable with its used alphabet size. Further, by definition, the entropy of the abstract stationary distribution, $H(\rho_\phi)$, gives us a lower bound on the number of bits needed to represent the used parts of $\mathcal{S}_\phi$. In this way, the entropy as a measure of compression is exploiting the fact that the most probable state can be written as `0`, the second most probable state as `10`, and so on. Thus, a lower entropy is already indicative of compressing $|\mathcal{S}_\phi|$. Further, in experiments, we find this upper bound is loose relative to the size of the abstract state space our algorithm produces.

Next, we introduce a second lemma that relates the expected KL divergence between two policies to the difference in value achieved by the policies, in expectation under some state distribution:

Lemma 2 *Consider two stochastic policies, π_1 and π_2 on state space $\mathcal{S}$, and a fixed probability distribution over $\mathcal{S}$, $p(s)$. If, for some $k \in \mathbb{R}_{\geq 0}$:*

$$\mathop{\mathbb{E}}_{p(s)} \left[D_{KL}(\pi_1(a \mid s) \mid\mid \pi_2(a \mid s)) \right] \leq k, \tag{16}$$

then:

$$\mathop{\mathbb{E}}_{p(s)} \left[V^{\pi_1}(s) - V^{\pi_2}(s) \right] \leq \sqrt{2k} \text{VMAX}, \tag{17}$$

where VMAX *is an upper bound on the value-function.*

The above bound lets us relate the distortion measure governed by IB to that of the CVA objective. Notably, this bound is vacuous for values of $k \geq \frac{1}{2}$.

The optimization problem presented by $\hat{\mathcal{J}}_{\text{DIB}}$ can be solved by the usual IB method. We thus introduce Determinstic Information Bottleneck for State abstractions (DIBS, presented in Algorithm 1), a simple iterative algorithm that adapts the DIB to Apprenticeship Learning with state abstractions. DIBS outputs a state-abstraction–policy pair in finite time that computes a local minimum of $\hat{\mathcal{J}}_{\text{DIB}}$, which we know from Theorem 2 is an upper bound on $\mathcal{J}$. The pseudocode presented is for the *deterministic* variant of the IB, as often state abstraction functions are treated as deterministic aggregation functions (Li, Walsh, and Littman 2006). The stochastic variant, which we call SIBS, will also be of interest, as soft state aggregation has been explored as well (Singh, Jaakkola, and Jordan 1995).

4 Experiments

We next describe two experiments that illustrate the power of DIBS for constructing abstractions that trade-off compression and value. First, we study the traditional Four Rooms domain introduced by Sutton, Precup, and Singh (1999). Second, we present a simple extension to SIBS that scales to high-dimensional observation spaces and evaluate this extension in the Atari game Breakout. Our code is made freely available for reproduction and extension.[1]

[1] github.com/david-abel/rl_info_theory

Algorithm 1 DIBS

INPUT: $\pi_E, \rho_E, M, \beta, \Delta$, *iters*
OUTPUT: ϕ, π_ϕ

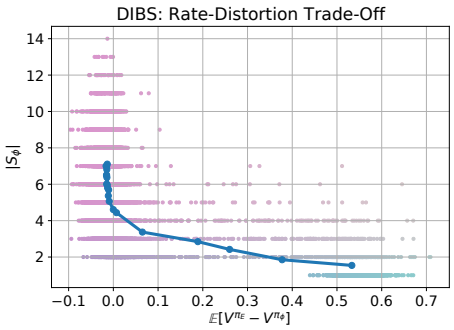
1: $\forall_s : \phi_0(s) = \texttt{random.choice}([1, |\mathcal{S}|])$ ▷ Initialize
2: $\forall_s : \pi_{\phi_0}(a \mid s_\phi) \sim \text{Unif}(\mathcal{A})$
3: $\forall_s : \rho_{\phi,0}(s_\phi) \sim \text{Unif}([1, |\mathcal{S}|])$
4: **for** $t = 0$ to *iters* **do** ▷ Iterative updates
5: $\quad j_{t+1}(s_\phi) = \log \rho_t(s_\phi) - \beta D_{KL}(\pi_E^s \mid\mid \pi_{\phi,t}^{s_\phi})$
6: $\quad \phi_{t+1}(s) = \arg\max_{s_\phi} j_{t+1}(s_\phi)$
7: $\quad \rho_{\phi,t+1}(s_\phi) = \sum_{s:\phi(s)=s_\phi} \rho_E(s)$
8: $\quad \pi_{\phi,t+1}(a \mid s_\phi) = \frac{\sum_{s:\phi(s)=s_\phi} \pi_E(a|s)\rho_E(s)}{\sum_{s:\phi(s)=s_\phi} \rho_E(s)}$
9: $\quad$ **if** $\max_{f \in \{\pi_\phi, \phi, \rho_\phi\}} \text{L}_1(f_t, f_{t+1}) \leq \Delta$ **then**
10: $\quad\quad$ `break` ▷ Converged
11: **return** $\phi_{t+1}, \pi_{\phi,t+1}$

4.1 Four Rooms

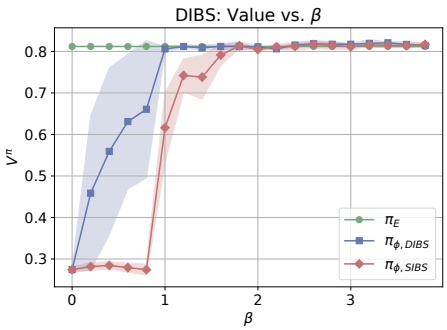
The first experiment focuses on the Four Rooms grid world domain pictured in 4a. The agent interacts with an 11×11 grid with walls dividing the world into four connected rooms. The agent has four actions, `up`, `left`, `down`, and `right`. Each action moves the agent in the specified direction with probability 0.9 (unless it hits a wall), and orthogonally with probability 0.05. The agent starts in the bottom left corner, and receives $+1$ reward for transitioning into the top right state, which is terminal. All other transitions receive 0 reward. We set γ to 0.99. The expert policy π_E is the optimal policy, with an additional $\varepsilon = 0.05$ probability of taking an action at random to ensure that an arbitrary stochastic policy over the action space has overlapping support with the expert policy.

In Four Rooms, we run DIBS and SIBS to convergence and compare the value of $\pi_{\phi,\text{DIBS}}$ and $\pi_{\phi,\text{SIBS}}$ to the value of the demonstrator policy for β between 0 and 4. We determine convergence as per line 9 of Algorithm 1: if all updating functions change by no more than Δ, the algorithm has converged. We set Δ to 0.001.

Figure 3a illustrates the rate-distortion trade-off made by 500 different runs of DIBS with different settings of β ranging from 0.0 to 4.0: each point indicates the size of the abstract state space (y-axis) and the value of the abstract policy relative to the demonstrator (x-axis), achieved by the computed state abstraction. Since the demonstrator is in fact suboptimal due to the ε-randomness, it is possible for the abstract policy to do slightly better than the demonstrator, as is the case for all points with x-value less than 0.0. The solid blue line corresponds to the average abstract state-space size and distortion achieved for different settings of β. When $\beta = 0$, DIBS prioritizes compression, yielding a one-state MDP on average (the far right blue point on the solid line). As β increases (which moves along the line to the left), the algorithm gradually tips the trade-off from prioritizing compression to prioritizing performance. When $\beta \geq 1.0$, we see the abstract policy achieve the same value as the demonstrator. Also of note is that a two-state MDP is capable of representing a policy (which could be stochastic) that is nearly

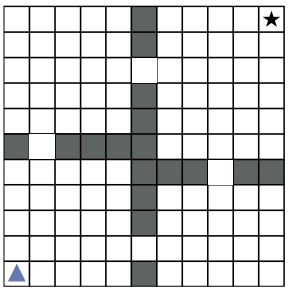


(a) RD Curve formed by DIBS.

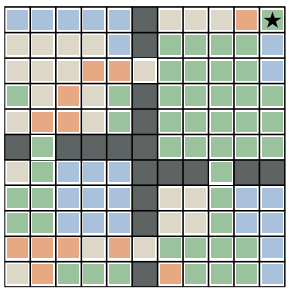


(b) Four Rooms: $V^\pi(s_0)$ vs. β

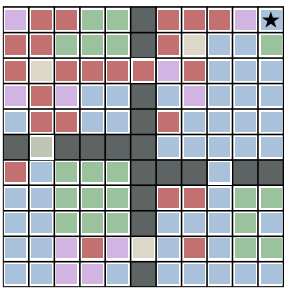
Figure 3: (a) The rate-distortion trade-off made by DIBS: each point indicates a single run of the algorithm on the Four Rooms domain. The x-axis corresponds to the sub-optimality of the abstract policy relative to the demonstrator, the y-axis denotes the size of the abstract state space. The blue line indicates the average, per choice of β between 0 and 4. (b) Illustrates the average value of the state-abstraction–policy combination (ϕ, π_ϕ) found by DIBS for different values of β.



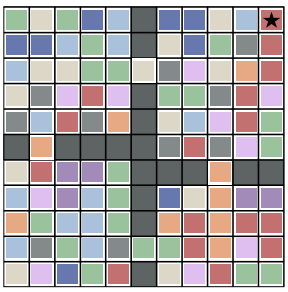
(a) The Four Rooms Domain



(b) ϕ with $\beta = 1$, $|\mathcal{S}_\phi| = 4$



(c) ϕ with $\beta = 2$, $|\mathcal{S}_\phi| = 5$



(d) ϕ with $\beta = 20$, $|\mathcal{S}_\phi| = 9$

Figure 4: (a) The Four Rooms domain from Sutton, Precup, and Singh (1999) and the state abstractions found by DIBS when (b) $\beta = 2$, (c) $\beta = 20$, and (d) $\beta = 200$. In (b), (c), and (d), cells of the same color are grouped into the same abstract state.

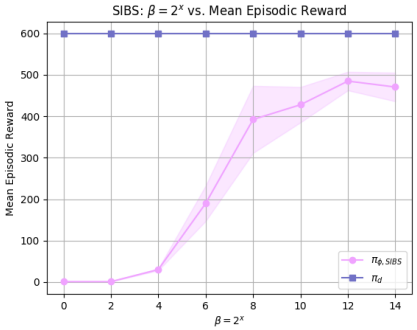
as effective as the expert policy. With a one state MDP, however, the best policies found by DIBS still result in around 0.45 value loss, relative to π_E.

Figure 3b offers a slightly different perspective on the same results. Here, we show the average value of the abstract policy achieved as a function of β, with β varying from 0 to 4.0. The results are averaged over 500 runs, with the lines indicating the average and shaded regions denoting 95% confidence intervals. Notably, when β is 0, we tend to find a policy that achieves significantly worse than the expert. With DIBS, as β increases, we see rapid improvement in the quality of the found policy, up until $\beta = 1$, at which point the abstract policy achieves almost identical performance to the expert. In contrast, the stochastic variant SIBS sees effectively no improvement in the quality of the policy until $\beta = 1$. We suspect this is due to the more difficult optimization problem presented, as the space of probabilistic state abstraction is much harder to search through than the space of deterministic ones. Still, as β increases past one, both $\pi_{\phi,\text{DIBS}}$ and $\pi_{\phi,\text{SIBS}}$ are able to almost exactly match the value of the demonstrator.

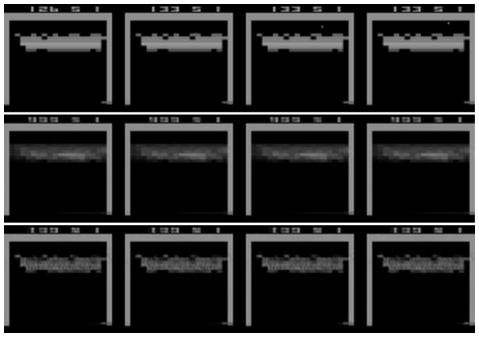
Figures 4b, 4c, and 4d show the state abstractions found by DIBS for $\beta = 1$, $\beta = 2$, and $\beta = 20$ respectively. Notably, each of these state abstractions were sufficient for effectively solving the problem. For the abstraction in Figure 4b, there are four abstract states, which is sufficient for nearly representing the demonstrator policy ("move up" in green, "move right" in tan, and so on). As we increase β, we find the abstraction yields far more states, though the quality of the optimal policy is already maxed out (there is simply less pressure to compress). In future work, we hope to explore the effect of this form of compression on generalization and sample complexity *beyond* just representation of the optimal policy. The intuition is that surely the abstract state spaces pictured in Figure 4b, 4c, and 4d will give rise to different degrees of learning difficulty, even though they can each represent a near-optimal policy. Exploring this question represents a major direction for future work.

4.2 Breakout

We next translate our algorithmic framework into domains with high-dimensional observations. To do so, we turn to variational autoencoders (VAEs) (Kingma and Welling 2013). In the VAE setting, we are concerned with learning a compact latent data representation, z, that captures a high-dimensional observation space, x, through the use of



(a) Breakout: Mean Reward vs. β



(b) Original (top) and ϕ with $\beta = 2$ (middle) and $\beta = 2048$ (bottom)

Figure 5: (a) The mean reward over 100 evaluation episodes of state-abstraction–policy (ϕ, π_ϕ) combinations found by our VAE-approximation to SIBS for different values of β, and (b) attempted state (a stack of four consecutive game screens) reconstructions using fixed state abstractions found when $\beta = 2$ and $\beta = 2048$ (b).

two parameterized functions $q_\psi(z \mid x)$ and $p_\theta(x \mid z)$. The pair represent a probabilistic encoder and decoder, typically captured by two separate neural networks, where the former maps data to a latent representation and the latter maps from z to the original observation. Traditionally, the two models are trained jointly to optimize the evidence lower bound objective (ELBO), which maximizes $\mathbb{E}_{q_\psi(z|x)}[\log p_\theta(x \mid z)]$ to facilitate reconstruction of the original data and minimizes $D_{\text{KL}}(q_\psi(z \mid x) \mid\mid p(z))$ to keep $q_\psi(z \mid x)$ close to some prior $p(z)$ over latent codes. Both $q_\psi(z \mid x)$ and $p(z)$ are commonly treated as Gaussian to use the Gaussian reparameterization trick (Kingma and Welling 2013). Since the ELBO is optimized in expectation over the data distribution, $p(x)$, we can leverage a known result regarding the KL divergence term (Kim and Mnih 2018):

$$\mathbb{E}_{p(x)}[D_{\text{KL}}(q_\psi(z \mid x) \mid\mid p(z))] = \\ I(X;Z) + D_{\text{KL}}(q(z) \mid\mid p(z)) \geq I(X;Z) \tag{18}$$

where $q(z) = \mathbb{E}_{p(x)}[q_\psi(z|x)]$. We treat x as the ground state representation $\mathcal{S}$ and z as the abstract state $\mathcal{S}_\phi$. We derive a new objective function that serves as a variational upper bound to the stochastic IB (SIBS) objective derived in Equation 12:

$$\min_\phi \mathop{\mathbb{E}}_{s\sim\rho_E}[D_{\text{KL}}(q_\psi(S_\phi \mid S) \mid\mid p(S_\phi)) + \\ \beta \mathop{\mathbb{E}}_{s_\phi\sim\phi(s)}[D_{\text{KL}}(\pi_E(a \mid s) \mid\mid \pi_\phi(a \mid s_\phi))]], \tag{19}$$

$$\geq \min_\phi I(S;S_\phi) + \\ \beta \mathop{\mathbb{E}}_{s\sim\rho_E, s_\phi\sim\phi(s)}[D_{KL}(\pi_E(a \mid s) \mid\mid \pi_\phi(a \mid s_\phi))], \tag{20}$$

where the upper bound follows directly from Equation 18.

To make use of this upper bound, we first create a demonstrator policy π_E for the Atari game Breakout (Bellemare et al. 2013) using A2C (Mnih et al. 2016). A Gaussian VAE agent is then trained with a separate architecture that has the same first four layers as the A2C agent before mapping out to a mean and covariance (in $\mathbb{R}^{25}$). Instead of reconstructing states, our decoder serves as an abstract policy network, mapping to a final distribution over the primitive actions, $\pi_\phi(a \mid s)$, which is really $\pi_\phi(a \mid z)$. The model is trained via Equation 19 for 2000 episodes using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.0001. During training, the expert's policy (π_E) controls the MDP.

We present results showcasing the effect of β on compression and performance in Figure 5; we find a relationship (mirroring that of Figure 3) between choice of β, success in approximating the demonstrator policy, and the nature of the resulting abstraction. In the visualizations of the abstraction, we observe that the quality of state reconstruction (each state is a row of four consecutive game screens) is compromised under a low setting of β (prioritizing compression), whereas a higher value of β preserves more information (paddle position and shape of bricks), leading to higher-quality reconstruction. Seeing that agent performance converges below the expert policy π_E, we suspect that increasing the size of the latent bottleneck could help close the gap.

5 Related Work

We now discuss connections to relevant work. Follow-up work on the Information Bottleneck explores application of the IB to RL. Rubin, Shamir, and Tishby (2012) introduce the control information of a policy for a given state, which leads to the trade-off between policy-information and policy value. Using dynamic programming, they compute the policy that makes this trade off. Tishby and Polani (2011) introduce the "information-to-go" function, which denotes the informational regret of choosing one policy over another. They then construct Bellman-like equations for deploying this quantity. Both of their methods and objectives are similar to ours, with one major departure: we consider the *abstractions* that achieve the appropriate trade-off, not policies, though the two are connected. Slonim and Tishby (2000) extends the IB to build a hierarchy of clusters on the input space such that each cluster can predict a target Y. We differ in that we focus on state abstraction for sequential decision making. We instead build our proxy objective from IB/DIB

to scale to more general settings, such as our VAE formulation and proposed extensions (see supplementary material).

Other prior work has investigated connections between information theory and RL. Serban et al. (2018) introduce the Bottleneck Simulator, a strategy for imposing capacity constraints to learn a factorized transition model, thereby achieving efficient model-based RL. Lerch and Sims (2018) introduce the Capacity Limited Actor-Critic (CLAC) based on RD theory, leading to an algorithm for capacity limited RL similar to Tishby and Polani (2011). Veness et al. (2015) study information theoretic methods for policy evaluation, while Still and Precup (2012) and Goyal et al. (2018) both leverage information theory for tackling the exploration problem. Lastly, Zhang and Wang (2018) offer a related spectral analysis of compression in Markov chains.

RD theory has been used in optimal control (Kostina et al. 2016; Ranade and Sahai 2015; Pandey and Kostina 2016). Our main departures consist of our formalism (RL vs. control), our use of a general, discrete transition matrix to model the environment (instead of linear-Gaussian dynamics), and our objective of maximizing value instead of properties like asymptotic system stability.

Further work has explored learning abstractions from a demonstrator (Mehta et al. 2007; Konidaris et al. 2010; Cobo et al. 2011; Le et al. 2018). We differ in our close attachment to both state abstraction and RD theory, through which we can explicitly trade-off between an abstraction's induced compression and representation of good behavior.

6 Conclusion

We articulated a new formalism for treating state abstraction as compression in Apprenticeship Learning. We introduced a new objective and proved that it is well-approximated by an IB-like objective, giving rise to a convergent algorithm for computing state abstractions that trade off between compression and value. Many questions remain. In the future, we first hope to explore extensions of our learning setting to cases wherein an expert policy is unavailable. We anticipate that a similar algorithmic scheme may be successfully applied to the traditional RL setting, wherein the agent controls the MDP. Second, as discussed previously, we hope to investigate the role that compression plays in ensuring low sample and planning complexity. Not all compressed models are the same: we hope to shed light on which kinds of compression actually lead to better generalization and faster learning. Lastly, we foresee extensions of our work to the lifelong or multitask setting, in which an RL agent must learn to solve a variety of related tasks. We discuss partial paths toward these directions in the supplementary material.

Acknowledgments

We thank George Konidaris, Stefanie Tellex, Milan Cvitkovic, and Ellis Hershkowitz for insightful discussions, and the anonymous 2016 ICML reviewer whose comments on our previous work helped inspire early versions of this project. Some support for this project was provided by NSF 1637614, and ONR MURI N00014-17-1-2699.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine learning*.

Abel, D.; Arumugam, D.; Lehnert, L.; and Littman, M. L. 2017. Toward good abstractions for lifelong learning. In *NeurIPS Workshop on Hierarchical Reinforcement Learning*.

Abel, D.; Arumugam, D.; Lehnert, L.; and Littman, M. L. 2018. State abstractions for lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 10–19.

Abel, D.; Hershkowitz, D. E.; and Littman, M. L. 2016. Near optimal behavior via approximate state abstraction. In *Proceedings of the International Conference on Machine Learning*.

Andre, D., and Russell, S. J. 2002. State abstraction for programmable reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 119–125. AAAI Press.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Arimoto, S. 1972. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory* 18(1):14–20.

Atkeson, C. G., and Schaal, S. 1997. Robot learning from demonstration. In *Proceedings of the International Conference on Machine Learning*.

Attneave, F. 1954. Some informational aspects of visual perception. *Psychological Review* 3(61).

Barlow, H. B. 1961. Possible principles underlying the transformations of sensory messages. *Sensory Communication*.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47:253–279.

Berger, T. 1971. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall.

Bertsekas, D. P., and Castanon, D. A. 1989. Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE transactions on Automatic Control* 34(6):589–598.

Blahut, R. 1972. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory* 18(4):460–473.

Botvinick, M.; Weinstein, A.; Solway, A.; and Barto, A. 2015. Reinforcement learning, efficient coding, and the statistics of natural tasks. *Current Opinion in Behavioral Sciences* 5:71–77.

Cobo, L. C.; Zang, P.; Isbell Jr, C. L.; and Thomaz, A. L. 2011. Automatic state abstraction from demonstration. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 22, 1243.

Dayan, P., and Hinton, G. E. 1993. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Dean, T.; Givan, R.; and Leach, S. 1997. Model reduction techniques for computing approximately optimal solutions for Markov decision processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 124–131.

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

Dietterich, T. G. 2000b. State abstraction in MAXQ hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 994–1000.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for finite Markov decision processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 162–169.

Givan, R.; Dean, T.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence* 147(1-2):163–223.

Goyal, A.; Islam, R.; Ahmed, Z.; Precup, D.; Botvinick, M.; Larochelle, H.; Levine, S.; and Bengio, Y. 2018. InfoBot: Structured exploration in reinforcement learning using information bottleneck. *ICML Workshop on Exploration in Reinforcement Learning*.

Hauskrecht, M.; Meuleau, N.; Kaelbling, L. P.; Dean, T.; and Boutilier, C. 1998. Hierarchical solution of Markov decision processes using macro-actions. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 220–229.

Hostetler, J.; Fern, A.; and Dietterich, T. G. 2014. State aggregation in monte carlo tree search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2446–2452. AAAI Press.

Jiang, N.; Singh, S.; and Lewis, R. 2014. Improving UCT planning via approximate homomorphisms. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Kim, H., and Mnih, A. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations*.

Konidaris, G.; Kuindersma, S.; Grupen, R.; and Barto, A. G. 2010. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in Neural Information Processing Systems*, 1162–1170.

Kostina, V.; Peres, Y.; Rácz, M. Z.; and Ranade, G. 2016. Rate-limited control of systems with uncertain gain. In *Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing*, 1189–1196. IEEE.

Le, H. M.; Jiang, N.; Agarwal, A.; Dudík, M.; Yue, Y.; and Daumé III, H. 2018. Hierarchical imitation and reinforcement learning. *Proceedings of the International Conference on Machine Learning*.

Lerch, R., and Sims, C. 2018. Exploration and policy generalization in capacity-limited reinforcement learning. *ICML Workshop on Exploration in RL*.

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*.

McCallum, A. K. 1996. *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. Dissertation, University of Rochester. Dept. of Computer Science.

Mehta, N.; Wynkoop, M.; Ray, S.; Tadepalli, P.; and Dietterich, T. G. 2007. Automatic induction of MAXQ hierarchies. In *NeurIPS Workshop on Hierarchical Organization of Behavior*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Mumey, B., and Gedeon, T. 2003. Optimal mutual information quantization is NP-complete. In *Neural Information Coding*.

Pandey, A., and Kostina, V. 2016. Information performance tradeoffs in control. *arXiv preprint arXiv:1611.01827*.

Puterman, M. L. 2014. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Ranade, G., and Sahai, A. 2015. Control capacity. In *International Symposium on Information Theory*. IEEE.

Rubin, J.; Shamir, O.; and Tishby, N. 2012. Trading value and information in MDPs. In *Decision Making with Imperfect Decision Makers*. Springer. 57–74.

Serban, I. V.; Sankar, C.; Pieper, M.; Pineau, J.; and Bengio, Y. 2018. The bottleneck simulator: A model-based deep reinforcement learning approach. *arXiv preprint arXiv:1807.04723*.

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27(3):379–423.

Sims, C. R. 2016. Rate–distortion theory and human perception. *Cognition* 152:181–198.

Sims, C. R. 2018. Efficient coding explains the universal law of generalization in human perception. *Science* 360(6389):652–656.

Singh, S. P.; Jaakkola, T.; and Jordan, M. I. 1995. Reinforcement learning with soft state aggregation. In *Advances in Neural Information Processing Systems*, 361–368.

Slonim, N., and Tishby, N. 2000. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems*.

Solway, A.; Diuk, C.; Córdova, N.; Yee, D.; Barto, A. G.; Niv, Y.; and Botvinick, M. M. 2014. Optimal behavioral hierarchy. *PLoS Computational Biology* 10(8):e1003779.

Still, S., and Precup, D. 2012. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences* 131(3):139–148.

Strouse, D., and Schwab, D. J. 2017. The deterministic information bottleneck. *Neural Computation* 29(6):1611–1630.

Sutter, T.; Sutter, D.; Esfahani, P. M.; and Lygeros, J. 2015. Efficient approximation of channel capacities. *IEEE Transactions on Information Theory* 61:1649–1666.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT press Cambridge, 2nd edition.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

Taïga, A. A.; Courville, A.; and Bellemare, M. G. 2018. Approximate exploration through state abstraction. *arXiv preprint arXiv:1808.09819*.

Tenenbaum, J. B.; Kemp, C.; Griffiths, T. L.; and Goodman, N. D. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science* 331:1279–1285.

Tishby, N., and Polani, D. 2011. Information theory of decisions and actions. In *Perception-Action Cycle*. Springer. 601–636.

Tishby, N.; Pereira, F. C.; and Bialek, W. 1999. The information bottleneck method. *The 37th Annual Allerton Conference on Communication, Control, and Computing*.

Veness, J.; Bellemare, M. G.; Hutter, M.; Chua, A.; and Desjardins, G. 2015. Compress and control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3016–3023.

Whitt, W. 1978. Approximations of dynamic programs, i. *Mathematics of Operations Research* 3(3):231–243.

Zhang, A., and Wang, M. 2018. State compression of Markov processes via empirical low-rank estimation. *arXiv preprint arXiv:1802.02920*.