Between Rate-Distortion Theory & Value Equivalence in Model-Based Reinforcement Learning

Dilip Arumugam
Department of Computer Science
Stanford University
dilip@cs.stanford.edu

Benjamin Van Roy
Department of Electrical Engineering
Department of Management Science & Engineering
Stanford University
bvr@stanford.edu

Abstract

The quintessential model-based reinforcement-learning agent iteratively refines its estimates or prior beliefs about the true underlying model of the environment. Recent empirical successes in model-based reinforcement learning with function approximation, however, eschew the true model in favor of a surrogate that, while ignoring various facets of the environment, still facilitates effective planning over behaviors. Recently formalized as the value equivalence principle, this algorithmic technique is perhaps unavoidable as real-world reinforcement learning demands consideration of a simple, computationally-bounded agent interacting with an overwhelmingly complex environment. In this work, we entertain an extreme scenario wherein some combination of immense environment complexity and limited agent capacity entirely precludes identifying an exactly value-equivalent model. In light of this, we embrace a notion of approximate value equivalence and introduce an algorithm for incrementally synthesizing *simple* and *useful* approximations of the environment from which an agent might still recover near-optimal behavior. Crucially, we recognize the information-theoretic nature of this lossy environment compression problem and use the appropriate tools of rate-distortion theory to make mathematically precise how value equivalence can lend tractability to otherwise intractable sequential decision-making problems.

Keywords: Bayesian reinforcement learning, Information theory, Model-based reinforcement learning, Efficient exploration

Acknowledgements

The authors gratefully acknowledge Christopher Grimm for initial discussions that provided an impetus for this work. Financial support from Army Research Office (ARO) grant W911NF2010055 is gratefully acknowledged.

1 Problem Formulation

We formulate a sequential decision-making problem as an episodic, finite-horizon Markov Decision Process (MDP) [4, 14] defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \beta, H \rangle$. $\mathcal{S}$ denotes a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is a deterministic reward function providing evaluative feedback signals (in the unit interval) to the agent, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is a transition function prescribing distributions over next states, $\beta \in \Delta(\mathcal{S})$ is an initial state distribution, and $H \in \mathbb{N}$ is the maximum episode length or horizon.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. As is standard in Bayesian reinforcement learning, both the transition function and reward function are not known to the agent and are consequently treated as random variables. With all other MDP components known a priori, the randomness in the model fully accounts for the randomness in the MDP, which is also a random variable. We denote by $\mathcal{M}^\star$ the true MDP with model $(\mathcal{R}^\star, \mathcal{T}^\star)$ that the agent interacts with and attempts to solve over the course of K episodes. Within each episode, the agent acts for exactly H steps beginning with an initial state $s_1 \sim \beta$. For each $h \in [H]$, the agent observes the current state $s_h \in \mathcal{S}$, selects action $a_h \sim \pi_h(\cdot \mid s_h) \in \mathcal{A}$, enjoys a reward $r_h = \mathcal{R}(s_h, a_h) \in [0, 1]$, and transitions to the next state $s_{h+1} \sim \mathcal{T}(\cdot \mid s_h, a_h) \in \mathcal{S}$.

A stationary, stochastic policy for timestep $h \in [H]$, $\pi_h : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, encodes a pattern of behavior mapping individual states to distributions over possible actions. Letting $\{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$ denote the class of all stationary, stochastic policies, a non-stationary policy $\pi = (\pi_1, \ldots, \pi_H) \in \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}^H$ is a collection of exactly H stationary, stochastic policies whose overall performance in any MDP $\mathcal{M}$ at timestep $h \in [H]$ when starting at state $s \in \mathcal{S}$ and taking action $a \in \mathcal{A}$ is assessed by its associated action-value function $Q^\pi_{\mathcal{M},h}(s, a) = \mathbb{E}\left[\sum\limits_{h'=h}^{H} \mathcal{R}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a\right]$, where the expectation integrates over randomness in the action selections and transition dynamics. Taking the value function as $V^\pi_{\mathcal{M},h}(s) = \mathbb{E}_{a \sim \pi_h(\cdot|s)}\left[Q^\pi_{\mathcal{M},h}(s, a)\right]$, we define the optimal policy $\pi^\star = (\pi^\star_1, \pi^\star_2, \ldots, \pi^\star_H)$ as achieving supremal value $V^\star_{\mathcal{M},h}(s) = \sup\limits_{\pi \in \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}^H} V^\pi_{\mathcal{M},h}(s)$ for all $s \in \mathcal{S}$, $h \in [H]$. We let $\tau_k = (s^{(k)}_1, a^{(k)}_1, r^{(k)}_1, \ldots, s^{(k)}_H, a^{(k)}_H, r^{(k)}_H, s^{(k)}_{H+1})$ be a random variable denoting the trajectory experienced by the agent in the kth episode. Meanwhile, $H_k = \{\tau_1, \tau_2, \ldots, \tau_{k-1}\} \in \mathcal{H}_k$ is a random variable representing the entire history of the agent's interaction within the environment at the start of the kth episode. Abstractly, a reinforcement-learning algorithm is a sequence of non-stationary policies $(\pi^{(1)}, \ldots, \pi^{(K)})$ where, for each episode $k \in [K]$, $\pi^{(k)} : \mathcal{H}_k \rightarrow \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$ is a function of the current history H_k. We note that no further restrictions on the state-action space $\mathcal{S} \times \mathcal{A}$, such as finiteness, have been made; notably, through our use of information theory, our algorithm may operate on any finite-horizon, episodic MDP although we leave the question of how to practically instantiate our algorithm for concrete settings of interest to future work.

2 Rate-Distortion Theory

We here provide a brief, high-level overview of rate-distortion theory [17] and encourage readers to consult [7] for more details. A lossy compression problem consumes as input a fixed information source $\mathbb{P}(X \in \cdot)$ and a distortion function $d : \mathcal{X} \times \mathcal{Z} \rightarrow \mathbb{R}_{\geq 0}$ which quantifies the loss of fidelity by using a compression Z in place of the original X. Then, for any distortion threshold $D \in \mathbb{R}_{\geq 0}$, the rate-distortion function quantifies the fundamental limit of lossy compression as

$$\mathcal{R}(D) = \inf_{Z \in \Lambda} \mathbb{I}(X; Z) \triangleq \inf_{Z \in \Lambda} \mathbb{E}\left[D_{\mathrm{KL}}(\mathbb{P}(X \in \cdot \mid Z) \mid\mid \mathbb{P}(X \in \cdot))\right] \qquad \Lambda \triangleq \{Z : \Omega \rightarrow \mathcal{Z} \mid \mathbb{E}\left[d(X, Z)\right] \leq D\},$$

where $\mathbb{I}(X; Z)$ denotes the mutual information and the infimum is taken over all random variables Z that incur bounded expected distortion, $\mathbb{E}\left[d(X, Z)\right] \leq D$. Naturally, $\mathcal{R}(D)$ represents the minimum number of bits of information that must be retained from X in order to achieve this bounded expected loss of fidelity. In keeping with the previous problem formulation, which does not assume discrete random variables, we note that the rate-distortion function is well-defined for information source and channel output random variables taking values on abstract alphabets [8]. Moreover, the problem of computing the rate-distortion function along with the channel that achieves its infimum is well-studied and solved by the classic Blahut-Arimoto algorithm [6, 1], which is computationally feasible for discrete channel outputs.

Just as in past work that studies satisficing in multi-armed bandit problems [15, 2, 3], we use rate-distortion theory to formalize and identify a simplified MDP $\widetilde{\mathcal{M}}_k$ that the agent will attempt to learn over the course of each episode $k \in [K]$. The episode dependence arises from utilizing the agent's current beliefs over the true MDP $\mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_k)$ as an information source to be lossily compressed.

3 The Value Equivalence Principle

As outlined in the previous section, the second input for a well-specified lossy-compression problem is a distortion function prescribing non-negative real values to realizations of the information source and channel output random variables $(\mathcal{M}^\star, \widetilde{\mathcal{M}})$ that quantify the loss of fidelity incurred by using $\widetilde{\mathcal{M}}$ in lieu of $\mathcal{M}^\star$. To define this function, we will leverage an approximate notion of value equivalence [10, 11]. For any arbitrary MDP $\mathcal{M}$ with model $(\mathcal{R}, \mathcal{T})$ and any stationary, stochastic policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, define the Bellman operator $\mathcal{B}^\pi_{\mathcal{M}} : \{\mathcal{S} \rightarrow \mathbb{R}\} \rightarrow \{\mathcal{S} \rightarrow \mathbb{R}\}$ as follows: $\mathcal{B}^\pi_{\mathcal{M}} V(s) \triangleq \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\mathcal{R}(s,a) + \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \left[V(s')\right]\right]$. The Bellman operator is a foundational tool in dynamic-programming approaches to reinforcement learning [5] and gives rise to the classic Bellman equation: for any MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \beta, H\rangle$ and any non-stationary policy $\pi = (\pi_1, \ldots, \pi_H)$, the value functions induced by π satisfy $V^\pi_{\mathcal{M},h}(s) = \mathcal{B}^{\pi_h}_{\mathcal{M}} V^\pi_{\mathcal{M},h+1}(s)$, for all $h \in [H]$ and with $V^\pi_{\mathcal{M},H+1}(s) = 0, \forall s \in \mathcal{S}$.

For any two MDPs $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \beta, H\rangle$ and $\widehat{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \widehat{\mathcal{R}}, \widehat{\mathcal{T}}, \beta, H\rangle$, Grimm et al. [10] define a notion of equivalence between them despite their differing models. For any policy class $\Pi \subseteq \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$ and value function class $\mathcal{V} \subseteq \{\mathcal{S} \rightarrow \mathbb{R}\}$, $\mathcal{M}$ and $\widehat{\mathcal{M}}$ are value equivalent with respect to Π and $\mathcal{V}$ if and only if $\mathcal{B}^\pi_{\mathcal{M}} V = \mathcal{B}^\pi_{\widehat{\mathcal{M}}} V, \forall \pi \in \Pi, V \in \mathcal{V}$. In words, two different models are deemed value equivalent if they induce identical Bellman updates under any pair of policy and value function from $\Pi \times \mathcal{V}$. Grimm et al. [10] prove that when $\Pi = \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$ and $\mathcal{V} = \{\mathcal{S} \rightarrow \mathbb{R}\}$, the set of all exactly value-equivalent models is a singleton set containing only the true model of the environment. The key insight behind value equivalence, however, is that practical model-based reinforcement-learning algorithms need not be concerned with modeling every granular detail of the underlying environment and may, in fact, stand to benefit by optimizing an alternative criterion besides the traditional maximum-likelihood objective [18, 12, 16]. Indeed, by restricting focus to decreasing subsets of policies $\Pi \subset \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$ and value functions $\mathcal{V} \subset \{\mathcal{S} \rightarrow \mathbb{R}\}$, the space of exactly value-equivalent models is monotonically increasing.

For brevity, let $\mathfrak{R} \triangleq \{\mathcal{S} \times \mathcal{A} \rightarrow [0,1]\}$ and $\mathfrak{T} \triangleq \{\mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})\}$ denote the classes of all reward functions and transition functions, respectively. Recall that, with all uncertainty in $\mathcal{M}^\star$ entirely driven by its model, we may think of the support of $\mathcal{M}^\star$ as $\mathfrak{M} \triangleq \mathfrak{R} \times \mathfrak{T}$. We define a distortion function on pairs of MDPs $d : \mathfrak{M} \times \mathfrak{M} \rightarrow \mathbb{R}_{\geq 0}$ for any $\Pi \subseteq \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}, \mathcal{V} \subseteq \{\mathcal{S} \rightarrow \mathbb{R}\}$ as

$$d_{\Pi,\mathcal{V}}(\mathcal{M}, \widehat{\mathcal{M}}) = \sup_{\substack{\pi \in \Pi \\ V \in \mathcal{V}}} ||\mathcal{B}^\pi_{\mathcal{M}} V - \mathcal{B}^\pi_{\widehat{\mathcal{M}}} V||^2_\infty = \sup_{\substack{\pi \in \Pi \\ V \in \mathcal{V}}} \left(\max_{s \in \mathcal{S}} |\mathcal{B}^\pi_{\mathcal{M}} V(s) - \mathcal{B}^\pi_{\widehat{\mathcal{M}}} V(s)|\right)^2.$$

In words, $d_{\Pi,\mathcal{V}}$ is the supremal squared Bellman error between MDPs $\mathcal{M}$ and $\widehat{\mathcal{M}}$ across all states $s \in \mathcal{S}$ with respect to the policy class Π and value function class $\mathcal{V}$.

4 Value-Equivalent Sampling for Reinforcement Learning

By virtue of the previous two sections, we are now in a position to define the lossy compression problem that characterizes a MDP $\widetilde{\mathcal{M}}_k$ that the agent will endeavor to learn in each episode $k \in [K]$ instead of the true MDP $\mathcal{M}^\star$. For any $\Pi \subseteq \{\mathcal{S} \rightarrow \Delta(\mathcal{A})\}$; $\mathcal{V} \subseteq \{\mathcal{S} \rightarrow \mathbb{R}\}$; $k \in [K]$; and $D \geq 0$, we define the rate-distortion function

$$\mathcal{R}^{\Pi,\mathcal{V}}_k(D) = \inf_{\widetilde{\mathcal{M}} \in \Lambda} \mathbb{I}_k(\mathcal{M}^\star; \widetilde{\mathcal{M}}) \triangleq \inf_{\widetilde{\mathcal{M}} \in \Lambda} \mathbb{E}\left[D_{\mathrm{KL}}(\mathbb{P}(\mathcal{M}^\star \in \cdot \mid \widetilde{\mathcal{M}}, H_k) \,||\, \mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_k)) \mid H_k\right], \quad (1)$$

where $\Lambda \triangleq \left\{\widetilde{\mathcal{M}} : \Omega \rightarrow \mathfrak{M} \mid \mathbb{E}[d_{\Pi,\mathcal{V}}(\mathcal{M}^\star, \widetilde{\mathcal{M}}) \mid H_k] \leq D\right\}$. This rate-distortion function characterizes the fundamental limit of lossy MDP compression under our chosen distortion measure resulting in a channel that retains the minimum amount of information from the true MDP $\mathcal{M}^\star$ while yielding an approximately value-equivalent MDP in expectation. Observe that this distortion constraint is a notion of approximate value equivalence which collapses to the exact value equivalence of Grimm et al. [10] as $D \to 0$. Meanwhile, as $D \to \infty$, we accommodate a more aggressive compression of the true MDP $\mathcal{M}^\star$ resulting in less faithful Bellman updates.

Algorithm 1 Posterior Sampling for Reinforcement Learning (PSRL) [19]

```
Input: Prior $\mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_1)$
for $k \in [K]$ do
    Sample $M_k \sim \mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_k)$
    Get optimal policy $\pi^{(k)} = \pi^\star_{M_k}$
    Execute $\pi^{(k)}$ and get trajectory $\tau_k$
    Update history $H_{k+1} = H_k \cup \tau_k$
    Induce posterior $\mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_{k+1})$
end for
```

Algorithm 2 Value-equivalent Sampling for Reinforcement Learning (VSRL)

```
Input: Prior distribution $\mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_1)$, Distortion threshold $D \in \mathbb{R}_{\geq 0}$, Distortion function $d_{\Pi,\mathcal{V}} : \mathfrak{M} \times \mathfrak{M} \to \mathbb{R}_{\geq 0}$
for $k \in [K]$ do
    Compute channel $\mathbb{P}(\widetilde{\mathcal{M}}_k \in \cdot \mid \mathcal{M}^\star)$ achieving $\mathcal{R}_k^{\Pi,\mathcal{V}}(D)$ limit (Equation 1)
    Sample MDP $M^\star \sim \mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_k)$
    Sample compressed MDP $M_k \sim \mathbb{P}(\widetilde{\mathcal{M}}_k \in \cdot \mid \mathcal{M}^\star = M^\star)$
    Compute optimal policy $\pi^{(k)} = \pi^\star_{M_k}$
    Execute $\pi^{(k)}$ and observe trajectory $\tau_k$
    Update history $H_{k+1} = H_k \cup \tau_k$
    Induce posterior $\mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_{k+1})$
end for
```

A standard algorithm for our problem setting is widely known as Posterior Sampling for Reinforcement Learning (PSRL) [19, 13], which we present as Algorithm 1, while our Value-equivalent Sampling for Reinforcement Learning (VSRL) is given as Algorithm 2. The key distinction between them is that, at each episode $k \in [K]$, the latter takes the posterior sample $M^\star \sim \mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_k)$ and passes it through the channel that achieves the rate-distortion limit (Equation 1) at this episode to get the M_k whose optimal policy is executed in the environment.

5 Discussion

Example 1 (A Multi-Resolution MDP)**.** *For a large but finite $N \in \mathbb{N}$, consider a sequence of MDPs, $\{\mathcal{M}_n\}_{n\in[N]}$, which all share a common action space $\mathcal{A}$ but vary in state space ($\mathcal{S}_n$), reward function, and transition function. Moreover, for each $n \in [N]$, the rewards of the nth MDP are bounded in the interval $[0, \frac{1}{n}]$. An agent is confronted with the resulting product MDP, $\mathcal{M}$, defined on the state space $\mathcal{S}_1 \times \ldots \times \mathcal{S}_N$ with action space $\mathcal{A}$ and rewards summed across the N constituent reward functions. The transition function is defined such that each action $a \in \mathcal{A}$ is executed across all N MDPs simultaneously and the resulting individual transitions are composed to make a transition of $\mathcal{M}$. For any value of N, PSRL will persistently act to identify the transition and reward structure of all $\{\mathcal{M}_n\}_{n\in[N]}$.*

Example 1 presents a scenario where, as $N \uparrow \infty$, a complex environment retains a wealth of information, and yet, only a subset of that information may be within the agent's reach or even necessary for producing reasonably competent behavior. VSRL implicitly identifies a $M \ll N$ such that learning the subsequence of MDPs $\{\mathcal{M}_n\}_{n\in[M]}$ is sufficient for achieving a desired degree of sub-optimality.

The core impetus for this work is to recognize that, for complex environments, pursuit of the exact MDP $\mathcal{M}^\star$ may be an entirely infeasible goal. Consider a MDP that represents control of a real-world, physical system; learning a transition function of the associated environment, at some level, demands that the agent internalize laws of physics and motion to a reasonable degree of accuracy. More formally, take the random variable $M_1 \sim \mathbb{P}(\mathcal{M}^\star \in \cdot \mid H_1)$ reflecting the agent's prior beliefs over $\mathcal{M}^\star$. Denoting $\mathbb{H}(\cdot)$ as the entropy of a random variable, observe that identifying $\mathcal{M}^\star$ requires that a PSRL agent obtain exactly $\mathbb{H}(M_1)$ bits of information from the environment which, under an uninformative prior, may either be prohibitively large and exceed the agent's capacity constraints or simply be impractical under time and resource constraints.

6 Conclusion

In this work, we embrace the idea of *satisficing* [15, 2, 3]; as succinctly stated by Herbert A. Simon during his 1978 Nobel Memorial Lecture, "decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world." Rather than spend an inordinate amount of time trying to recover an optimum solution to the true environment, VSRL pursues optimum solutions for a sequence of simplified environments. Future work will develop a complementary regret analysis that demonstrates how finding such optimum solutions for simplified worlds ultimately acts as a mechanism for achieving a satisfactory solution for the realistic, complex world. Naturally, the loss of fidelity between the simplified and true environments translates into a fixed amount of regret that an agent designer consciously and willingly accepts for two reasons: (1) they expect a reduction in the amount of time, data, and bits of information needed to identify the simplified environment and (2) in tasks where the environment encodes irrelevant information and exact knowledge isn't needed to achieve optimal behavior [9, 10, 11], a VSRL agent may still identify the optimal policy while maintaining greater sample efficiency than traditional PSRL.

References

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

[2] Dilip Arumugam and Benjamin Van Roy. Deciding what to learn: A rate-distortion approach. In *International Conference on Machine Learning*, pages 373–382. PMLR, 2021.

[3] Dilip Arumugam and Benjamin Van Roy. The value of information when deciding what to learn. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

[5] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[6] Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, 1972.

[7] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

[8] Imre Csiszár. On an extremum problem of information theory. *Studia Scientiarum Mathematicarum Hungarica*, 9, 1974.

[9] Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

[10] Christopher Grimm, Andre Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[11] Christopher Grimm, Andre Barreto, Gregory Farquhar, David Silver, and Satinder Singh. Proper value equivalence. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[12] Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6120–6130, 2017.

[13] Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pages 2701–2710. PMLR, 2017.

[14] Martin L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.

[15] Daniel Russo and Benjamin Van Roy. Satisficing in time-sensitive bandit learning. *Mathematics of Operations Research*, 2022.

[16] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, et al. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[17] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec., March 1959*, 4:142–163, 1959.

[18] David Silver, Hado Hasselt, Matteo Hessel, et al. The Predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, pages 3191–3199. PMLR, 2017.

[19] Malcolm JA Strens. A Bayesian framework for reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 943–950, 2000.