

Mutual-Information Regularization in Markov Decision Processes and Actor-Critic Learning

Felix Leibfried, Jordi Grau-Moya
PROWLER.io
Cambridge, UK
{felix,jordi}@prowler.io

Abstract: Cumulative entropy regularization introduces a regulatory signal to the reinforcement learning (RL) problem that encourages policies with high-entropy actions, which is equivalent to enforcing small deviations from a uniform reference marginal policy. This has been shown to improve exploration and robustness, and it tackles the value overestimation problem. It also leads to a significant performance increase in tabular and high-dimensional settings, as demonstrated via algorithms such as soft Q-learning (SQL) and soft actor-critic (SAC). Cumulative entropy regularization has been extended to optimize over the reference marginal policy instead of keeping it fixed, yielding a regularization that minimizes the mutual information between states and actions. While this has been initially proposed for Markov Decision Processes (MDPs) in tabular settings, it was recently shown that a similar principle leads to significant improvements over vanilla SQL in RL for high-dimensional domains with *discrete* actions and function approximators.

Here, we follow the motivation of mutual-information regularization from an inference perspective and theoretically analyze the corresponding Bellman operator. Inspired by this Bellman operator, we devise a novel mutual-information regularized actor-critic learning (MIRACLE) algorithm for *continuous* action spaces that optimizes over the reference marginal policy. We empirically validate MIRACLE in the Mujoco robotics simulator, where we demonstrate that it can compete with contemporary RL methods. Most notably, it can improve over the model-free state-of-the-art SAC algorithm which implicitly assumes a fixed reference policy.

Keywords: Mutual-Information Regularization, MDP, Actor-Critic Learning

1 Introduction

In RL and MDPs, agents aim at collecting maximum reward yielding optimal policies that are *deterministic*. One way of obtaining *non-deterministic* optimal policies is by introducing a regulatory signal that penalizes deviations from a stochastic reference policy [1, 2, 3, 4, 5, 6]. For a uniform reference policy, cumulative entropy regularization is recovered [7, 8, 9, 10, 11, 12]. The effect of the latter on RL is: improved exploration and robustness, and an alleviation of overestimated values [3, 10, 6]. Importantly, overall performance is improved not only in the tabular setting [3], but also in high-dimensional settings with function approximators as demonstrated via algorithms such as SQL and SAC [5, 10, 6]. In [13, 14], it has been proposed to optimize over the reference policy rather than keeping it fixed, yielding a constrained reward maximization problem with a constraint on the mutual information between states and actions in each time step. This has been investigated in the tabular setting by [13] for MDPs, and in high-dimensional domains and RL by [14] where it was shown that for discrete actions, mutual-information regularization can lead to significant improvements over a fixed reference policy (i.e. SQL [5, 6]). Here, we follow the RL-as-inference perspective where optimizing for the reference policy increases the log marginal likelihood. Our contributions are threefold. A) we theoretically analyze the mutual-information-regularized Bellman operator. B) we then develop a mutual-information regularized actor-critic learning (MIRACLE) algorithm for continuous action spaces based on this Bellman operator. And C) we demonstrate that MIRACLE can attain results competitive with contemporary methods in challenging large-scale robotics domains from Mujoco—e.g. improving over the state-of-the-art SAC [10] in Ant.

2 Background

In the following, we provide some background and notation for MDPs and RL, see Section 2.1. Subsequently in Section 2.2, we focus on mutual-information regularization in non-sequential decision-making scenarios paving the way for the sequential decision-making setting discussed in Section 3.

2.1 Markov Decision Process (MDP) and Reinforcement Learning (RL)

An MDP is a five-tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where \mathcal{S} is the state set and \mathcal{A} the action set. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0; 1]$ refers to the probabilistic state-transition function that maps state-action pairs $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$ to next states $\mathbf{s}' \in \mathcal{S}$ according to the conditional probability distribution $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$. The reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ determines the instantaneous reward $R(\mathbf{s}, \mathbf{a})$ when taking action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$. The hyperparameter $\gamma \in (0; 1)$ is a discount factor for discounting future rewards. In the MDP setting, an agent is specified by a behavioral policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$ that maps states to actions probabilistically according to the conditional probability distribution $\pi(\mathbf{a}|\mathbf{s})$. A policy π is evaluated based on its expected future cumulative reward captured by the state value function V^π and the state-action value function Q^π respectively as:

$$V^\pi(\mathbf{s}) := \mathbb{E}_{\pi, P} \left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \middle| \mathbf{s}_0 = \mathbf{s} \right] \quad \text{and} \quad Q^\pi(\mathbf{s}, \mathbf{a}) := R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V^\pi(\mathbf{s}')], \quad (1)$$

where t is the time index and \mathbf{s}' refers to the next state. The goal is to identify *optimal* policies π^* that maximize expected future cumulative reward, yielding optimal value functions V^* and Q^* as:

$$V^*(\mathbf{s}) := \max_{\pi} \mathbb{E}_{\pi, P} \left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \middle| \mathbf{s}_0 = \mathbf{s} \right] \quad \text{and} \quad Q^*(\mathbf{s}, \mathbf{a}) := R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V^*(\mathbf{s}')]. \quad (2)$$

In the RL setting, agents do not have prior knowledge about the transition function P and the reward function R , and need to identify optimal policies via environment interactions. A popular class of RL algorithms that are suited for high-dimensional environments and continuous action spaces are actor-critic algorithms that learn a parametric policy π_ϕ with policy parameters ϕ as:

$$\max_{\phi} \mathbb{E}_{p, \pi_\phi} [Q^{\pi_\phi}(\mathbf{s}, \mathbf{a})], \quad (3)$$

where Q^{π_ϕ} is the state-action value function of π_ϕ and $p : \mathcal{S} \rightarrow [0; 1]$ is a distribution over states. Since Q^{π_ϕ} can usually not be computed in closed form, it is approximated via function approximators, policy rollouts or a combination of both. The state distribution p refers either to an empirical distribution induced by a replay memory that collects visited states [15, 16, 17, 18, 10, 12] or to the stationary state distribution when executing π_ϕ in the environment [19, 20, 21].

2.2 Mutual-Information Regularization for Non-Sequential Decision-Making

A non-sequential decision-making scenario can be described via a four-tuple $(\mathcal{S}, \mathcal{A}, R, p)$ comprising a state set \mathcal{S} , an action set \mathcal{A} , a reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and a state distribution $p : \mathcal{S} \rightarrow [0; 1]$ from which states are sampled according to $\mathbf{s} \sim p(\mathbf{s})$. A behavioral policy is specified the same way as in the sequential setting as $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$ yielding a conditional probability distribution $\pi(\mathbf{a}|\mathbf{s})$ that maps states $\mathbf{s} \in \mathcal{S}$ to actions $\mathbf{a} \in \mathcal{A}$. The mutual-information regularized decision-making problem is then defined in its constrained and unconstrained form respectively as:

$$\max_{\pi} \mathbb{E}_{p, \pi} [R(\mathbf{s}, \mathbf{a})] \quad \text{s.t.} \quad I(\mathbf{S}, \mathbf{A}) \leq C \quad \iff \quad \max_{\pi} \mathbb{E}_{p, \pi} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a}|\mathbf{s})}{\pi_{\text{marginal}}(\mathbf{a})} \right], \quad (4)$$

where $C > 0$ is an upper bound on the mutual information between states and actions $I(\mathbf{S}, \mathbf{A})$ and $\pi_{\text{marginal}}(\mathbf{a}) = \sum_{\mathbf{s}} p(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})$ is the marginal action distribution. β steers the trade-off between reward maximization and mutual information minimization in the unconstrained problem description.

The intuition behind the above objective is to interpret an agent as an information-theoretic channel $\pi(\mathbf{a}|\mathbf{s})$ with input \mathbf{s} and output \mathbf{a} . The agent aims at reconstructing the input at the output as described by the function $R(\mathbf{s}, \mathbf{a})$. Perfect reconstruction would generally incur high mutual information between input and output but the channel capacity $I(\mathbf{S}, \mathbf{A})$ is upper-bounded via C .

This requires therefore the agent to discard such information in \mathbf{s} that has little impact on R in order to not violate the information constraint. The framework is equivalent to rate distortion theory from information theory [22, 23] (a special case of the information bottleneck [24]) and is versatile in the sense that it applies to a wide range of decision-making systems. In the past, it has been applied to different scientific fields such as: A) economics—where it is referred to as *rational inattention* [25]—describing humans as bounded-rational decision-makers; B) information-theoretic decision making to explain the emergence of abstract representations as a consequence of limited information-processing capabilities [23]; C) theoretical neuroscience to develop biologically plausible weight update rules for spiking neurons that prevent synaptic growth without bounds [26]; and D) machine learning where it translates into a regularizer improving generalization in deep neural networks in classification tasks [27].

3 Mutual-Information Regularization in MDPs

In the next two sections, we first formulate the problem of mutual-information regularization in MDPs from an inference perspective following [11, 14]—see Section 3.1. We then proceed by providing a theoretical analysis of the corresponding Bellman operator in Section 3.2.

3.1 Problem Formulation from an Inference Perspective

In statistical inference, the goal is to infer the distribution of some latent variables given observations. The connection between latent and observable variables is determined through a generative model that specifies a prior distribution over latent variables and a likelihood of observable variables conditioned on latent variables. Interestingly, the RL problem of maximizing expected cumulative reward can be phrased as an inference problem by specifying latents and observables in a certain way [11] as outlined in the following. Assuming an undiscounted finite horizon problem with horizon T , latents are specified as the state-action trajectory $\tau = \mathbf{s}_0, \mathbf{a}_0, \dots, \mathbf{s}_T, \mathbf{a}_T$ distributed according to the prior distribution $p(\tau) = p_{\text{ini}}(\mathbf{s}_0)\pi_{\text{prior}}(\mathbf{a}_0) \prod_{t=1}^T P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1})\pi_{\text{prior}}(\mathbf{a}_t)$ where $p_{\text{ini}} : \mathcal{S} \rightarrow [0; 1]$ is an initial state distribution and $\pi_{\text{prior}} : \mathcal{A} \rightarrow [0; 1]$ is a prior action distribution that is state-unconditioned. A single observable is specified artificially as a binary variable r distributed according to the likelihood $p(r = 1|\tau) = \frac{1}{Z} \exp\left(\beta \sum_{t=0}^T R(\mathbf{s}_t, \mathbf{a}_t)\right)$ where Z is a normalization constant and $\beta > 0$ a scaling factor. The inference goal is to identify the posterior distribution $p(\tau|r = 1) = \frac{p(r=1|\tau)p(\tau)}{\sum_{\tau} p(r=1|\tau)p(\tau)}$ over trajectories given the observation $r = 1$. Since computing this posterior is in general intractable, common practice is to resort to variational inference and approximate it via a variational distribution $q(\tau) = p_{\text{ini}}(\mathbf{s}_0)\pi(\mathbf{a}_0|\mathbf{s}_0) \prod_{t=1}^T P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_{t-1})\pi(\mathbf{a}_t|\mathbf{s}_t)$ that is parameterized via the policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$, a state-conditioned probability distribution. An optimal q implied by an optimal policy π is then identified through maximizing a lower bound of the log marginal likelihood $\log \sum_{\tau} p(r = 1|\tau)p(\tau)$ referred to as the evidence lower bound (ELBO):

$$\begin{aligned} \log \sum_{\tau} p(r = 1|\tau)p(\tau) &\geq \max_q \sum_{\tau} q(\tau) \left(\log p(r = 1|\tau) - \log \frac{q(\tau)}{p(\tau)} \right) \\ &\equiv \max_{\pi} \mathbb{E}_{p_{\text{ini}}, \pi, P} \left[\sum_{t=0}^T R(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\text{prior}}(\mathbf{a}_t)} \right], \end{aligned} \quad (5)$$

where the ‘ \equiv ’-sign is because of re-scaling by $\frac{1}{\beta}$ and ignoring the normalization constant Z in $p(r = 1|\tau)$ since it only adds a constant offset to the objective. Equation (5) demonstrates how to recover the RL problem under a soft policy constraint [7, 1, 3, 8, 9, 4, 5, 10, 6, 12] from an inference perspective. More details on the subject can be found for example in [11].

Common practice in contemporary variational inference methods is to optimize the ELBO not only w.r.t. to the variational distribution q but also w.r.t. aspects of the generative model itself [28]—for example the prior—in order to obtain a better log marginal likelihood. Following this reasoning suggests to *optimize* over the prior action distribution π_{prior} rather than keeping it fixed, leading to

$$\max_{\pi, \pi_{\text{prior}}} \mathbb{E}_{p_{\text{ini}}, \pi, P} \left[\sum_{t=0}^T R(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\text{prior}}(\mathbf{a}_t)} \right], \quad (6)$$

which is similar to Equation (5) except for optimizing over the state-unconditioned prior as well.

Notice how Equation (6) resembles the non-sequential decision-making formulation with mutual-information regularization from Equation (4) in Section 2.2. What remains to be understood is how the incurred logarithmic penalty signal relates to a mutual information constraint. We proceed in line with [14] by expressing the unnormalized marginal state distribution $p^\pi(\mathbf{s})$ of the policy π as

$$p^\pi(\mathbf{s}) = p_{\text{ini}}(\mathbf{s}) + \sum_{t=1}^T p_t^\pi(\mathbf{s}) \text{ with } p_t^\pi(\mathbf{s}) = \sum_{\mathbf{s}_0, \dots, \mathbf{a}_{t-1}} p_{\text{ini}}(\mathbf{s}_0) \prod_{t'=1}^t \pi(\mathbf{a}_{t'-1} | \mathbf{s}_{t'-1}) P(\mathbf{s}_{t'} | \mathbf{s}_{t'-1}, \mathbf{a}_{t'-1}), \quad (7)$$

where $\mathbf{s} \hat{=} \mathbf{s}_t$ for $p_t^\pi(\mathbf{s})$. Optimizing Equation (6) w.r.t. π_{prior} for a fixed π can then be formulated as

$$\max_{\pi_{\text{prior}}} \mathbb{E}_{p^\pi, \pi} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a} | \mathbf{s})}{\pi_{\text{prior}}(\mathbf{a})} \right] \iff \min_{\pi_{\text{prior}}} \mathbb{E}_{p^\pi} [D_{KL}(\pi(\cdot | \mathbf{s}) \| \pi_{\text{prior}}(\cdot))], \quad (8)$$

which corresponds to minimizing the expected Kullback-Leibler divergence between the policy π and the prior π_{prior} averaged over the unnormalized marginal state distribution p^π . It is known that the optimal solution to this problem is just the marginal policy $\pi_{\text{prior}}^*(\mathbf{a}) = \sum_{\mathbf{s}} \pi(\mathbf{a} | \mathbf{s}) p_{\text{norm}}^\pi(\mathbf{s}) = \pi_{\text{marginal}}(\mathbf{a})$ where p_{norm}^π refers to the normalized marginal state distribution [22, 14]. Replacing π_{prior} in the expected D_{KL} -term in Equation (8) with π_{marginal} yields the unnormalized mutual information $I(\mathcal{S}, \mathcal{A})$ scaled by a factor as a consequence of the unnormalized marginal state distribution. Utilizing the marginal policy π_{marginal} , we can express Equation (6) finally as follows

$$\max_{\pi} \mathbb{E}_{p^\pi, \pi} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a} | \mathbf{s})}{\pi_{\text{marginal}}(\mathbf{a})} \right] \iff \max_{\pi} \mathbb{E}_{p^\pi, \pi} [R(\mathbf{s}, \mathbf{a})] \text{ s.t. } I(\mathcal{S}, \mathcal{A}) \leq C, \quad (9)$$

in its unconstrained and constrained form respectively bridging the gap to the non-sequential formulation in Equation (4) from Section 2.2. Importantly, Equation (9) is also valid in a discounted infinite horizon setting where $p^\pi(\mathbf{s}) = p_{\text{ini}}(\mathbf{s}) + \sum_{t=1}^{\infty} \gamma^t p_t^\pi(\mathbf{s})$ as explained for example in [14].

A notable difference compared to the non-sequential setting is that the unnormalized marginal state distribution p^π depends on the optimization argument π , which renders the optimization problem non-trivial. It has therefore been suggested to introduce a policy-independent state distribution $p : \mathcal{S} \rightarrow [0; 1]$ [13] for computing the marginal policy π_{marginal} —stronger consistency assumptions on p that adhere to state transitions have not been investigated so far and are left for future work. For infinite horizon scenarios, we then obtain the following Bellman operator:

$$B_\star V(\mathbf{s}) := \max_{\pi} \mathbb{E}_{\pi} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a} | \mathbf{s})}{\pi_{\text{prior}}(\mathbf{a})} + \gamma \mathbb{E}_P [V(\mathbf{s}')] \right] \text{ s.t. } \pi_{\text{prior}}(\mathbf{a}) = \sum_{\mathbf{s}} \pi(\mathbf{a} | \mathbf{s}) p(\mathbf{s}), \quad (10)$$

where the constrained problem formulation highlights that the optimal prior π_{prior}^* is the true marginal action distribution $\pi_{\text{marginal}}(\mathbf{a}) = \sum_{\mathbf{s}} \pi(\mathbf{a} | \mathbf{s}) p(\mathbf{s})$ as outlined earlier. The marginal action distribution however depends on the behavioral policy in all states, not just the one that occurs above on the l.h.s. of the equation in $B_\star V(\mathbf{s})$. This leads to the problem that the Bellman operator $B_\star : \mathbb{R}^{\mathcal{S}} \rightarrow \mathbb{R}^{\mathcal{S}}$ cannot be applied independently to all states $\mathbf{s} \in \mathcal{S}$ as in ordinary value iteration schemes. The consequence of the latter is that standard theoretical tools for analyzing unique optimal values and convergence cannot be applied straightforwardly [29, 2, 30], although there is empirical evidence that mutual-information regularized value iteration can converge in grid world examples [13].

Apart from the difficulty of analyzing the overall convergence behaviour of mutual-information regularized value iteration, applying the actual Bellman operator B_\star from Equation (10) is also not trivial since there is no closed-form solution for an optimal policy-prior pair. In the next section (Section 3.2), we therefore provide a theoretical analysis on how to *apply* the operator B_\star to a given value function $V : \mathcal{S} \rightarrow \mathbb{R}$. This results in a practical algorithm that iteratively re-computes the optimal ‘one-step’ policy and the optimal ‘one-step’ prior w.r.t. the current value estimates in an alternate fashion until convergence. In Section 4, we devise a novel actor-critic algorithm inspired by Equation (10) for continuous action spaces, and demonstrate that it can attain competitive performance with state-of-the-art algorithms in the robotics simulation domain of Mujoco.

Remark 1 *In a broader sense, the formulation above relates to other sequential decision-making formulations based on the information bottleneck [31] and information-processing hierarchies [32].*

3.2 Theoretical Analysis of the Bellman Operator

Our theoretical contribution (Theorem 1) is an iterative algorithm to apply the Bellman operator B_* .

Theorem 1 *On the Application of the Bellman Operator B_* . Under the assumption that the reward function R is bounded, the optimization problem in Equation (10) imposed by the Bellman operator B_* can be solved by iterating in an alternate fashion through the following two equations:*

$$\pi_{\text{prior}}^{(m)}(\mathbf{a}) = \sum_{\mathbf{s}} \pi^{(m)}(\mathbf{a}|\mathbf{s})p(\mathbf{s}), \quad (11)$$

$$\pi^{(m+1)}(\mathbf{a}|\mathbf{s}) = \frac{\pi_{\text{prior}}^{(m)}(\mathbf{a})e^{\beta(R(\mathbf{s},\mathbf{a})+\gamma\sum_{\mathbf{s}'}P(\mathbf{s}'|\mathbf{s},\mathbf{a})V(\mathbf{s}'))}}{\sum_{\mathbf{a}}\pi_{\text{prior}}^{(m)}(\mathbf{a})e^{\beta(R(\mathbf{s},\mathbf{a})+\gamma\sum_{\mathbf{s}'}P(\mathbf{s}'|\mathbf{s},\mathbf{a})V(\mathbf{s}'))}}, \quad (12)$$

where m refers to the iteration index. Denoting the total number of iterations as M , the presented scheme converges at a rate of $O(1/M)$ to an optimal policy π^* for any given bounded value function V and any initial policy $\pi^{(0)} : \mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$ that has support in $\mathcal{A} \forall \mathbf{s} \in \mathcal{S}$. Such a scheme is commonly referred to as Blahut-Arimoto-type algorithm [22].

Proof. The problem is similar to rate distortion theory [22] from information theory and so is identifying an optimal solution, accomplished via Lemmas 1 and 2 and Proposition 1 next. Lemma 1 deals with how to compute an optimal prior given a fixed policy, while Lemma 2 deals with how to compute an optimal policy given a fixed prior. This leads to a set of self-consistent equations whose alternate application converges to an optimal solution as proven by Proposition 1. \square

Definition 1 *Evaluation Operator. A specific prior-policy-pair $(\pi_{\text{prior}}, \pi)$ is evaluated as*

$$B_{\pi_{\text{prior}}, \pi} V(\mathbf{s}) := \mathbb{E}_{\pi} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}(\mathbf{a})} + \gamma \mathbb{E}_P [V(\mathbf{s}')] \right]. \quad (13)$$

Lemma 1 *Optimal Prior for a Given Policy. The optimal prior for Equation (10) given a bounded reward function R , a bounded value function V and a policy π is:*

$$\arg \max_{\pi_{\text{prior}}} \mathbb{E}_p [B_{\pi_{\text{prior}}, \pi} V(\mathbf{s})] = \sum_{\mathbf{s}} \pi(\mathbf{a}|\mathbf{s})p(\mathbf{s}). \quad (14)$$

Proof. Maximizing the expected evaluation operator $\mathbb{E}_p [B_{\pi_{\text{prior}}, \pi} V(\mathbf{s})]$ w.r.t. π_{prior} is equivalent to minimizing the expected Kullback-Leibler divergence $\mathbb{E}_p [D_{KL}(\pi(\cdot|\mathbf{s})||\pi_{\text{prior}}(\cdot))]$, because the reward and the value function do not depend on the prior. It holds that $I(\mathcal{S}, \mathcal{A}) \leq \mathbb{E}_p [D_{KL}(\pi(\cdot|\mathbf{s})||\pi_{\text{prior}}(\cdot))]$ for all π_{prior} [22], implying that the optimal prior is the true marginal action distribution averaged over all states. \square

Lemma 2 *Optimal Policy for a Given Prior. The optimal policy for Equation (10) given a bounded reward function R , a bounded value function V and a prior policy π_{prior} is:*

$$\arg \max_{\pi} B_{\pi_{\text{prior}}, \pi} V(\mathbf{s}) = \frac{\pi_{\text{prior}}(\mathbf{a})e^{\beta(R(\mathbf{s},\mathbf{a})+\gamma\sum_{\mathbf{s}'}P(\mathbf{s}'|\mathbf{s},\mathbf{a})V(\mathbf{s}'))}}{\sum_{\mathbf{a}}\pi_{\text{prior}}(\mathbf{a})e^{\beta(R(\mathbf{s},\mathbf{a})+\gamma\sum_{\mathbf{s}'}P(\mathbf{s}'|\mathbf{s},\mathbf{a})V(\mathbf{s}'))}}. \quad (15)$$

Proof. First note that for a given prior π_{prior} , Equation (10) can be solved independently for each state $\mathbf{s} \in \mathcal{S}$. Identifying an optimal policy then becomes an optimization problem subject to the constraint $\sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}) = 1$, which can be straightforwardly solved with the method of Lagrange multipliers and standard variational calculus yielding Equation (15)—see Appendix A in line with [23]. \square

Corollary 1 *Concise Bellman Operator. For a specific prior-policy-pair $(\pi_{\text{prior}}^{(m)}, \pi^{(m+1)})$ obtained by running the Blahut-Arimoto scheme for m rounds according to Equations (11) and (12), it holds:*

$$B_{\pi_{\text{prior}}^{(m)}, \pi^{(m+1)}} V(\mathbf{s}) = \frac{1}{\beta} \log \mathbb{E}_{\pi_{\text{prior}}^{(m)}} \left[e^{\beta(R(\mathbf{s},\mathbf{a})+\gamma\mathbb{E}_P[V(\mathbf{s}'))]} \right], \quad (16)$$

which is obtained by plugging Equation (12) into $B_{\pi_{\text{prior}}, \pi} V(\mathbf{s})$ from Equation (13) under the assumption of having the ‘compatible’ prior $\pi_{\text{prior}} = \pi_{\text{prior}}^{(m)}$.

Proposition 1 *Convergence.* Given a bounded reward function R , a bounded value function V and an initial policy $\pi^{(0)}$ with support in $\mathcal{A} \forall \mathbf{s} \in \mathcal{S}$, iterating through Equations (11) and (12) converges to an optimal policy π^* at a rate of $O(1/M)$ where M is the total number of iterations.

Proof. The proof is accomplished via Lemmas 3 and 4 in line with [33]. We first specify an upper bound (Lemma 3) for the solution to the optimization problem in Equation (10), i.e. the expected Bellman operator averaged over states, which we use to complete the proof in Lemma 4. \square

Lemma 3 *Upper Expected Optimal Value Bound.* The solution to Equation (10) is upper-bounded:

$$\mathbb{E}_p [B_\star V(\mathbf{s})] \leq \frac{1}{\beta} \mathbb{E}_{p, \pi^\star} \left[\log \frac{\pi^{(m+1)}(\mathbf{a}|\mathbf{s})}{\pi^{(m)}(\mathbf{a}|\mathbf{s})} \right] + \mathbb{E}_p \left[B_{\pi_{\text{prior}}^{(m)}, \pi^{(m+1)}} V(\mathbf{s}) \right], \quad (17)$$

where the superscript \star refers to an optimal solution, and the superscripts (m) and $(m+1)$ to preliminary solutions after m and $m+1$ iterations of the Blahut-Arimoto scheme respectively.

Proof. The key ingredient for deriving the upper bound in Equation (17) is to show that for all m :

$$\mathbb{E}_p [B_\star V(\mathbf{s})] \leq \mathbb{E}_{p, \pi^\star} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi^{(m)}(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}^{(m)}(\mathbf{a})} + \gamma \mathbb{E}_P [V(\mathbf{s}')] \right], \quad (18)$$

which we detail in the first part of Appendix B with help of [34]. Using Equation (12) for $\pi^{(m+1)}$ and Corollary 1, we then obtain Equation (17) as detailed in the second part of Appendix B. \square

Lemma 4 *Completing Convergence.* The convergence proof is completed by showing that

$$\frac{1}{M} \sum_{m=0}^{M-1} \left(\mathbb{E}_p [B_\star V(\mathbf{s})] - \mathbb{E}_p \left[B_{\pi_{\text{prior}}^{(m)}, \pi^{(m+1)}} V(\mathbf{s}) \right] \right) \leq \frac{1}{M\beta} \mathbb{E}_p \left[\max_{\mathbf{a}} \log \left(\pi^{(0)}(\mathbf{a}|\mathbf{s}) \right)^{-1} \right], \quad (19)$$

where $\pi^{(0)}$ is the initial policy at the start of the Blahut-Arimoto algorithm, implying a rate $O(1/M)$.

Proof. This is obtained by rearranging Equation (17) such that only the logarithmic policy term remains on the right-hand side. Taking the average over M iterations and swapping the expectation over p and π^\star with the sum on the right-hand side, all the logarithmic policy terms except for M and 0 cancel. Under worst case assumptions, one arrives at Equation (19)—details in Appendix C. \square

In Figure 1, we validate B_\star in a grid world confirming that mutual-information regularized value iteration can converge in line with [13], and compare against soft values where the prior is fixed.

Remark 2 Corollary 1 shows that the mutual-information regularized Bellman operator generalizes the soft Bellman operator [1, 2, 3, 4, 5, 6], and cumulative entropy regularization as a further special case of the latter [7, 8, 9, 10, 11, 12], when fixing the prior and not optimizing it.

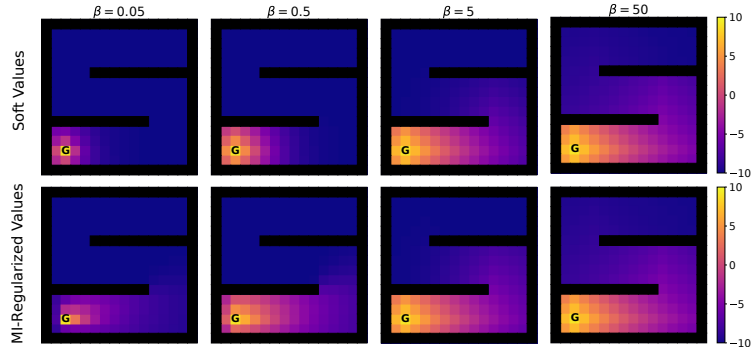


Figure 1: Effect of β on State Values. Soft values (uniform prior) and mutual-information-regularized values are compared. For small β , soft values evaluate the prior while mutual-information-regularized values are optimal for a state-independent policy—details in Appendix E.

4 Mutual-Information Regularization with Continuous Actions

While the previous section (Section 3.2) deals with a theoretical analysis of the Bellman operator B_* resulting from mutual-information regularization, we devise here a mutual-information regularized actor-critic learning (MIRACLE) algorithm inspired by B_* that scales to high dimensions and can handle continuous action spaces. Our hypothesis is that optimizing the ELBO from Equation (5) w.r.t. the prior π_{prior} should lead to a better log marginal likelihood [28] and hence to better performance, as empirically verified with deep parametric function approximators in domains with discrete actions such as Atari [14]. In the next section (Section 5), we empirically validate our method in the robotics simulation domain Mujoco and demonstrate competitive performance with contemporary methods. Specifically, we show that optimizing over the prior can improve over the state-of-the-art soft actor-critic (SAC) algorithm [10, 12] that implicitly assumes a fixed uniform prior.

Following the policy gradient theorem [15] and in line with Equation (3), MIRACLE optimizes the parameters ϕ of a parametric policy $\pi_\phi : \mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$ by gradient ascent on the objective:

$$J_\phi = \mathbb{E}_{\mathcal{D}, \pi_\phi} \left[Q_\theta(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi_\phi(\mathbf{a}|\mathbf{s})}{\pi_\chi(\mathbf{a})} \right], \quad (20)$$

where states are sampled from an empirical non-stationary distribution, i.e. a replay buffer \mathcal{D} that collects environment interactions of the agent [16, 17, 18, 10], and actions from the policy π_ϕ . $Q_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and $\pi_\chi : \mathcal{A} \rightarrow [0; 1]$ are additional function approximators with parameters θ and χ that approximate the Q-values of the policy π_ϕ and the marginal policy of π_ϕ averaged over states respectively. The logarithmic term that penalizes deviations of the policy π_ϕ from its marginal π_χ is a direct consequence of mutual-information regularization, where we leverage on the fact that the optimal prior state-unconditioned policy is the true marginal action distribution—see Section 3.1.

To learn Q-values efficiently, we follow [10] and introduce an additional V-critic $V_\psi : \mathcal{S} \rightarrow \mathbb{R}$ parameterized by ψ to approximate state values of the policy π_ϕ . Optimal parameters θ of the Q-critic are then learned via gradient descent on the following average squared loss:

$$J_\theta = \mathbb{E}_{\mathcal{D}} \left[(Q_\theta(\mathbf{s}, \mathbf{a}) - R(\mathbf{s}, \mathbf{a}) - \gamma V_\psi(\mathbf{s}'))^2 \right], \quad (21)$$

where \mathcal{D} refers to a distribution that samples state-action-next-state tuples from the replay buffer.

Optimal parameters of the V-critic are trained via gradient descent on the following loss:

$$J_\psi = \mathbb{E}_{\mathcal{D}} \left[\left(V_\psi(\mathbf{s}) - \mathbb{E}_{\pi_\phi} \left[Q_\theta(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi_\phi(\mathbf{a}|\mathbf{s})}{\pi_\chi(\mathbf{a})} \right] \right)^2 \right], \quad (22)$$

where the replay buffer \mathcal{D} is used to sample states and the policy π_ϕ to sample actions.

Finally, one needs to approximate the (state-unconditioned) marginal action distribution π_χ of the policy π_ϕ . Abusing notation, we learn a probabilistic parametric map $\pi_\chi : \mathbb{R}^{\dim(\mathcal{A})} \times \mathcal{A} \rightarrow [0; 1]$, i.e. a conditional probability distribution $\pi_\chi(\mathbf{a}|\mathbf{u})$, that maps samples $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{0}_{\dim(\mathcal{A})}, \mathbf{I}_{\dim(\mathcal{A})})$ from the standard Gaussian to actions $\mathbf{a} \in \mathcal{A}$ from the replay buffer \mathcal{D} . This can be phrased as a supervised problem and achieved with max log likelihood. The parametric marginal action distribution is then: $\pi_\chi(\mathbf{a}) = \int_{\mathbf{u}} \pi_\chi(\mathbf{a}|\mathbf{u}) \mathcal{N}(\mathbf{u}|\mathbf{0}_{\dim(\mathcal{A})}, \mathbf{I}_{\dim(\mathcal{A})}) d\mathbf{u}$ and can be approximated as $\pi_\chi(\mathbf{a}) \approx \frac{1}{N} \sum_{n=1}^N \pi_\chi(\mathbf{a}|\mathbf{u}_n)$ to estimate probability values in Equations (20) and (22), where \mathbf{u}_n are i.i.d. Gaussian samples. A fixed state distribution, as in our theoretical analysis in Section 3.2 and in [13] for tabular environments, would be non-sensible in Mujoco because of the problem of admissible states. The marginal action distribution marginalizes therefore over states from the replay buffer instead to approximate the policy-induced marginal action distribution as outlined in Section 3.1 and in line with [14]. Experiments using a variational autoencoder with a recognition model [35] to approximate the marginal policy did not yield better results—see Appendix F and G.

The training procedure for MIRACLE is off-policy and similar to [10] where the agent interacts with the environment iteratively and stores state-action-reward-next-state tuples in a replay buffer. After every step, a minibatch of transition tuples is sampled from the buffer to perform a single gradient step on the objectives for the different learning components of the algorithm. We also practically operate with a reward scale, i.e. we scale the reward function by β rather than the logarithmic penalty term by its inverse β^{-1} [10]. During training, we apply the reparameterization trick [35, 36, 10] for π_ϕ and π_χ in their respective objectives—details can be found in Appendix C of [10] explaining how to reparameterize when action spaces are bounded (as in Mujoco).

5 Experiments in Mujoco

We empirically validate MIRACLE on the latest v2-environments of Mujoco. Function approximators are parameterized with deep nets trained with Adam [37]. We use a twin Q-critic [38, 10] and limit the variance of the probabilistic policy networks π_ϕ and π_χ following [39]. Further details can be found in Appendix F. Our baselines are DDPG [16] and PPO [21] from RLlib [40] as well as SAC [10]. Note that we are mainly concerned comparing to the latter since we introduce a marginal reference policy to be optimized over compared to an implicit fixed uniform marginal policy in SAC—we therefore use an implementation for SAC and MIRACLE that only differs in this aspect and is the same otherwise. Every experiment is conducted with *ten* seeds.

Figure 2 shows that MIRACLE consistently outperforms SAC on lower-dimensional environments. On high-dimensional environments, MIRACLE is better than DDPG and PPO, and can outperform SAC. However, MIRACLE might not always help since it favours actions that are close to past actions, which may differ from actions that are identified by the optimization procedure [14].

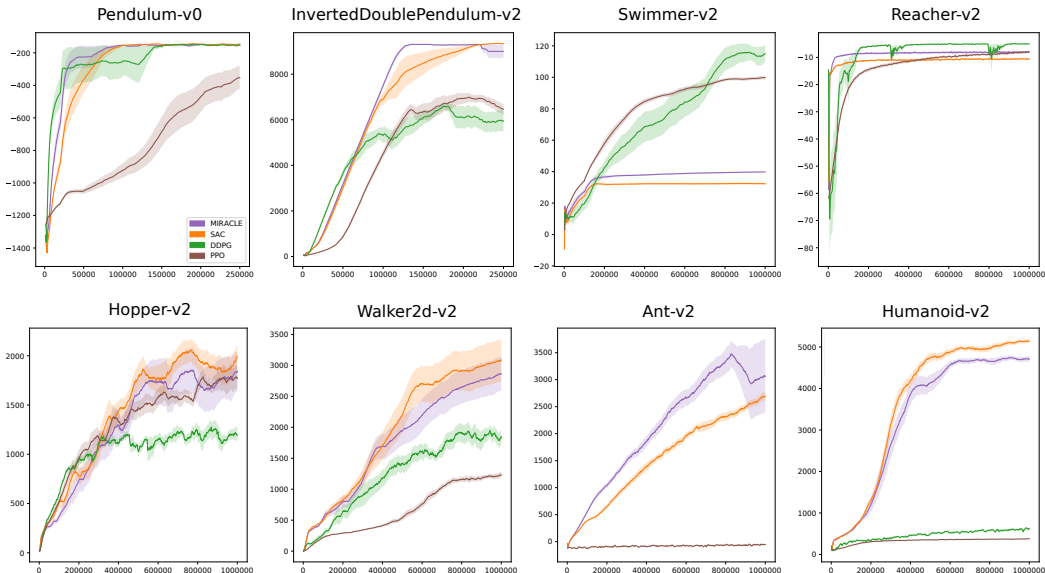


Figure 2: Mujoco Experiments. The figure reports average episodic rewards over the last 100 episodes and standard errors. MIRACLE consistently outperforms SAC in lower-dimensional environments (top row). In high-dimensional tasks, MIRACLE outperforms DDPG and PPO, and can outperform SAC significantly, see Ant (bottom row). In general, MIRACLE may not always help since it encourages the policy to stay close to past actions, which can be different to actions identified by the optimization procedure in line with experiments in discrete-action environments [14]. In Ant (kink caused by one seed), DDPG from RLlib did not work properly in our setup—see Appendix G.

6 Conclusion

Motivating mutual-information-regularization in MDPs from an inference perspective leads to a Bellman operator that generalizes the soft Bellman operator from the literature. We provide a theoretical analysis of this operator resulting in a practically applicable algorithm. Inspired by that, we devise an actor-critic algorithm (with an adaptive marginal prior) for high-dimensional continuous domains and demonstrate competitive results compared to contemporary methods in Mujoco, e.g. improvements in Ant over the state-of-the-art SAC (with an implicit fixed marginal prior).

Acknowledgments

We thank Joshua Aduol for helpful suggestions on the engineering side.

References

- [1] M. G. Azar, V. Gomez, and H. J. Kappen. Dynamic policy programming with function approximation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011.
- [2] J. Rubin, O. Shamir, and N. Tishby. Trading value and information in MDPs. In *Decision Making with Imperfect Decision Makers*, chapter 3. Springer, 2012.
- [3] R. Fox, A. Pakman, and N. Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.
- [4] G. Neu, V. Gomez, and A. Jonsson. A unified view of entropy-regularized Markov decision processes. *arXiv*, 2017.
- [5] J. Schulman, P. Abbeel, and X. Chen. Equivalence between policy gradients and soft Q-learning. *arXiv*, 2017.
- [6] F. Leibfried, J. Grau-Moya, and H. Bou-Ammar. An information-theoretic optimality principle for deep reinforcement learning. In *NIPS Workshop*, 2018.
- [7] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Carnegie Mellon University, USA, 2010.
- [8] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning*, 2017.
- [9] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in Neural Information Processing Systems*, 2017.
- [10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*, 2018.
- [11] S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv*, 2018.
- [12] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *arXiv*, 2019.
- [13] N. Tishby and D. Polani. Information theory of decisions and actions. In *Perception-Action Cycle*, chapter 19. Springer, 2011.
- [14] J. Grau-Moya, F. Leibfried, and P. Vrancx. Soft Q-learning with mutual-information regularization. In *Proceedings of the International Conference on Learning Representations*, 2019.
- [15] T. Degris, M. White, and R. S. Sutton. Off-policy actor-critic. In *Proceedings of the International Conference on Machine Learning*, 2012.
- [16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2016.
- [17] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. In *Proceedings of the International Conference on Learning Representations*, 2018.
- [18] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning*, 2018.
- [19] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

- [20] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2015.
- [21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. In *arXiv*, 2017.
- [22] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley & Sons, 2006.
- [23] T. Genewein, F. Leibfried, J. Grau-Moya, and D. A. Braun. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. *Frontiers in Robotics and AI*, 2(27), 2015.
- [24] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, 1999.
- [25] C. A. Sims. Implications of rational inattention. *Journal of Monetary Economics*, 50(3):665–690, 2003.
- [26] F. Leibfried and D. A. Braun. A reward-maximizing spiking neuron as a bounded rational decision maker. *Neural Computation*, 27(8):1686–1720, 2015.
- [27] F. Leibfried and D. A. Braun. Bounded rational decision-making in feedforward neural networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.
- [28] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.
- [29] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Springer, 1996.
- [30] J. Grau-Moya, F. Leibfried, T. Genewein, and D. A. Braun. Planning with information-processing constraints and model uncertainty in Markov decision processes. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2016.
- [31] A. Goyal, R. Islam, D. J. Strouse, Z. Ahmend, H. Larochelle, M. Botvinick, Y. Bengio, and S. Levine. InfoBot: Transfer and exploration via the information bottleneck. In *Proceedings of the International Conference on Learning Representations*, 2019.
- [32] H. Hihn, S. Gottwald, and D. A. Braun. An information-theoretic on-line learning principle for specialization in hierarchical decision-making systems. *arXiv*, 2019.
- [33] R. G. Gallager. The Arimoto-Blahut algorithm for finding channel capacity. Technical report, Massachusetts Institute of Technology, USA, 1994.
- [34] Y. Polyanskiy and Y. Wu. Chapter 4: Extremization of mutual information: Capacity saddle point. In *Massachusetts Institute of Technology Lecture Notes: Information Theory*, 2016.
- [35] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.
- [36] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*, 2014.
- [37] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.
- [38] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.
- [39] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.
- [40] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica. RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018.

A Details Regarding the Proof of Lemma 2

Formulating the Lagrangian for Lemma 2 yields:

$$L(\pi, \lambda) = \mathbb{E}_\pi \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}(\mathbf{a})} + \gamma \mathbb{E}_P [V(\mathbf{s}')] \right] - \lambda \left(\left(\sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}) \right) - 1 \right). \quad (23)$$

Taking the derivative of $L(\pi, \lambda)$ w.r.t. to π for a specific action, equating with zero and resolving for π leads to:

$$\pi_{\text{prior}}(\mathbf{a}) e^{\beta(R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V(\mathbf{s}')]) - 1 - \beta\lambda}. \quad (24)$$

Taking the derivative of $L(\pi, \lambda)$ w.r.t. to λ , plugging in Equation (24), equating with zero and resolving for $\exp(-1 - \beta\lambda)$, one arrives at:

$$e^{-1 - \beta\lambda} = \left(\sum_{\mathbf{a}} \pi_{\text{prior}}(\mathbf{a}) e^{\beta(R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V(\mathbf{s}')])} \right)^{-1}. \quad (25)$$

Plugging Equation (25) back into Equation (24), one arrives at Equation (15) from the main paper.

B Details Regarding the Proof of Lemma 3

First, we detail how to obtain Equation (18). This is accomplished by using the inequality $\mathbb{E}_p [D_{KL}(\pi^*(\cdot|\mathbf{s}) \|\pi^{(m)}(\cdot|\mathbf{s}))] \geq D_{KL}(\pi_{\text{prior}}^*(\cdot) \|\pi_{\text{prior}}^{(m)}(\cdot))$ —known as ‘conditioning increases divergence’, see e.g. [34]. Rearranging then yields: $\mathbb{E}_{p, \pi^*} \left[-\log \frac{\pi^*(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}^*(\mathbf{a})} \right] \leq \mathbb{E}_{p, \pi^*} \left[-\log \frac{\pi^{(m)}(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}^{(m)}(\mathbf{a})} \right]$, which leads to Equation (18) because rewards and values are policy-independent. The upper value bound in Equation (17) from Lemma 3 is then derived starting from Equation (18) as follows:

$$\begin{aligned} \mathbb{E}_p [B_\star V(\mathbf{s})] &\leq \mathbb{E}_{p, \pi^*} \left[R(\mathbf{s}, \mathbf{a}) - \frac{1}{\beta} \log \frac{\pi^{(m)}(\mathbf{a}|\mathbf{s})}{\pi_{\text{prior}}^{(m)}(\mathbf{a})} + \gamma \mathbb{E}_P [V(\mathbf{s}')] \right] \\ &= \mathbb{E}_{p, \pi^*} \left[\frac{1}{\beta} \log \left(e^{\beta R(\mathbf{s}, \mathbf{a}) + \log \pi_{\text{prior}}^{(m)}(\mathbf{a}) + \beta \gamma \mathbb{E}_P [V(\mathbf{s}')] } \right) - \frac{1}{\beta} \log \pi^{(m)}(\mathbf{a}|\mathbf{s}) \right] \\ &= \mathbb{E}_{p, \pi^*} \left[\frac{1}{\beta} \log \left(\pi_{\text{prior}}^{(m)}(\mathbf{a}) e^{\beta(R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V(\mathbf{s}')])} \right) - \frac{1}{\beta} \log \pi^{(m)}(\mathbf{a}|\mathbf{s}) \right] \\ &= \mathbb{E}_{p, \pi^*} \left[\frac{1}{\beta} \log \pi^{(m+1)}(\mathbf{a}|\mathbf{s}) - \frac{1}{\beta} \log \pi^{(m)}(\mathbf{a}|\mathbf{s}) \right] \\ &\quad + \mathbb{E}_p \left[\frac{1}{\beta} \log \mathbb{E}_{\pi_{\text{prior}}^{(m)}} \left[e^{\beta(R(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_P [V(\mathbf{s}')])} \right] \right] \\ &= \frac{1}{\beta} \mathbb{E}_{p, \pi^*} \left[\log \frac{\pi^{(m+1)}(\mathbf{a}|\mathbf{s})}{\pi^{(m)}(\mathbf{a}|\mathbf{s})} \right] + \mathbb{E}_p \left[B_{\pi_{\text{prior}}^{(m)}, \pi^{(m+1)}} V(\mathbf{s}) \right]. \end{aligned} \quad (26)$$

To obtain the second to last line, we use Equation (12) for $\pi^{(m+1)}$, and to obtain the last line, we use Equation (16) from Corollary 1.

C Details Regarding the Proof of Lemma 4

Rearranging Equation (17) and averaging over M iterations, one obtains the following:

$$\frac{1}{M} \sum_{m=0}^{M-1} \left(\mathbb{E}_p [B_\star V(\mathbf{s})] - \mathbb{E}_p \left[B_{\pi_{\text{prior}}^{(m)}, \pi^{(m+1)}} V(\mathbf{s}) \right] \right) \leq \frac{1}{M\beta} \mathbb{E}_{p, \pi^*} \left[\log \frac{\pi^{(M)}(\mathbf{a}|\mathbf{s})}{\pi^{(0)}(\mathbf{a}|\mathbf{s})} \right], \quad (27)$$

which leads directly to Equation (19) by taking the max over actions and using $\pi^{(M)}(\mathbf{a}|\mathbf{s}) \leq 1$.

D Remark Regarding Lemma 4

Lemma 4 also suggests to initialize $\pi^{(0)}$ with a uniform distribution $\mathcal{U}(\mathcal{A})$ since it minimizes the upper bound in Equation (19), more formally $\mathcal{U}(\mathcal{A}) = \arg \min_{\pi} \max_{\mathbf{a}} \log(\pi(\mathbf{a}))^{-1}$ for all admissible probability distributions π over the action set \mathcal{A} .

E Grid World Setup

In the grid world example from Figure 1 in Section 3.2 from the main paper, the agent has to reach a goal in the bottom left of a 16×16 grid. Reaching the goal is rewarded with +9 and terminates the episode, whereas each step is penalized with -1 . The agent can take five actions in each state, i.e. $\mathcal{A} = \{\text{'left'}, \text{'right'}, \text{'up'}, \text{'down'}, \text{'stay'}\}$. The environment is deterministic and the discount factor is $\gamma = 0.9$. The stopping criterion for the value iteration scheme is when the infinity norm of the difference value vector of two consecutive iterations drops below $5 \cdot 10^{-3}$ (for both soft value iteration as well as mutual-information-regularized value iteration). The stopping criterion for the inner Blahut-Arimoto scheme that is necessary in order to apply B_{\star} (required for one value iteration step) is when the maximum absolute difference in probability values in two consecutive inner iterations drops below $5 \cdot 10^{-3}$.

F Experiment Details for Mujoco

All function approximators are trained with the Adam optimizer [37] using a learning rate of $3 \cdot 10^{-4}$. The discount factor is $\gamma = 0.99$ and the replay buffer can store at most one million transition tuples. The minibatch size for training is 256 for all objectives. All deep networks have two hidden layers with 256 relu-activations implemented with PyTorch. When updating Q-values, we use an exponentially averaged V-target network with time parameter $\tau = 0.01$ [10]. The reward scale β is set to 10 for all experiments (also for the SAC baseline to ensure a fair comparison). Marginal policy values are approximated with 20 samples from the uniform Gaussian. We also use a separate replay buffer for training the marginal policy with buffer sizes depending on the environment (Pendulum: 10^3 ; InvertedDoublePendulum, Swimmer, Reacher: 10^4 ; Hopper, Walker2d, Ant, Humanoid: $5 \cdot 10^4$). Hyperparameters for the baselines DDPG, PPO and SAC are close to the literature [21, 18, 10] but with the same neural network architectures as above for a fair comparison.

The state value network V_{ψ} and the state-action value network Q_{θ} are ordinary feedforward networks that output scalar values. The policy network π_{ϕ} and the marginal policy network π_{χ} output a mean and a log standard deviation vector encoding a Gaussian. Since actions in Mujoco are bounded, unbounded actions are adjusted through a tanh-nonlinearity accordingly.

We also experimented with a proper variational autoencoder [35] to learn a generative model for marginal actions with an ELBO objective. The autoencoder requires an additional parametric recognition model—in our case a two-hidden-layer neural network (like the ones mentioned earlier) that maps actions to a mean and a log standard deviation vector representing a Gaussian in latent space.

G Experiment Results in Mujoco

The kink in performance of MIRACLE in Ant is due to the fact that in one out of 10 runs, episodic rewards significantly started dropping after around 800,000 to 900,000 steps, while in all the other runs, the policies kept improving. Also note that there is no DDPG baseline in Ant. In initial experiments on Ant, we observed that for our hyperparameter setting, the RLlib implementation immediately dropped to large negative reward values and never recovered from there.

Here, we also report in Figure 3 the maximum value obtained so far (in the course of training) of the average episodic reward over the last 100 episodes [39], as opposed to the main paper that reports the raw version of this metric. Reporting results with this metric highlights improvements of MIRACLE over baselines more clearly. Additional results similar to Figures 2 and 3 for experiments with variational autoencoders to model marginal actions are shown in Figures 4 and 5. These were conducted under the same setup as the other experiments and yielded worse results. Better results might be obtained by a different training procedure, e.g. more training samples for the autoencoder.

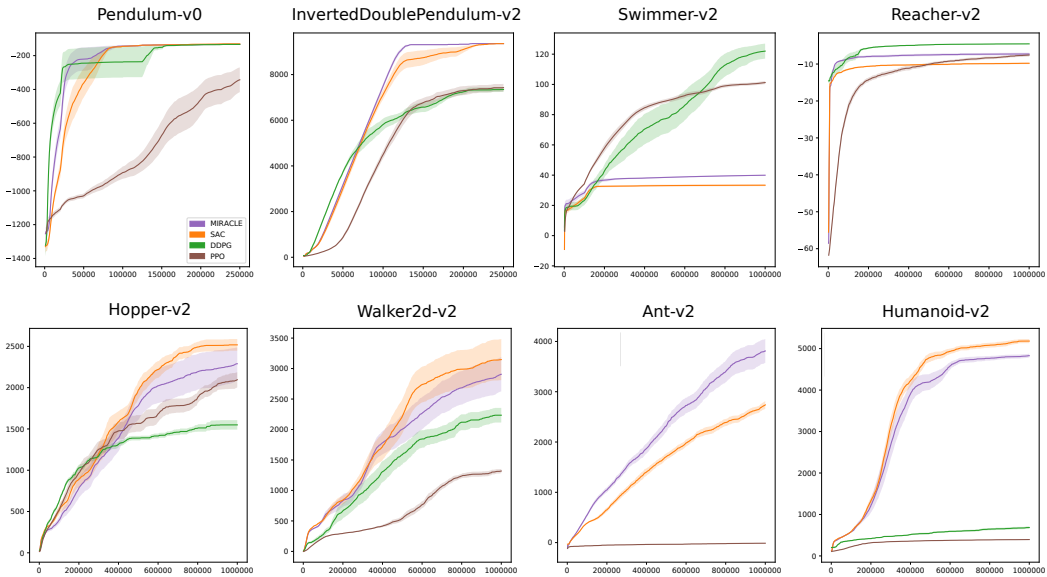


Figure 3: Mujoco Experiments Best Rewards. The plot shows the same experiments as in Figure 2 from the main paper, but reports the best episodic reward obtained so far during training (averaged over the last 100 episodes). Under this metric, MIRACLE clearly improves over SAC in five out of eight environments, most notably Ant.

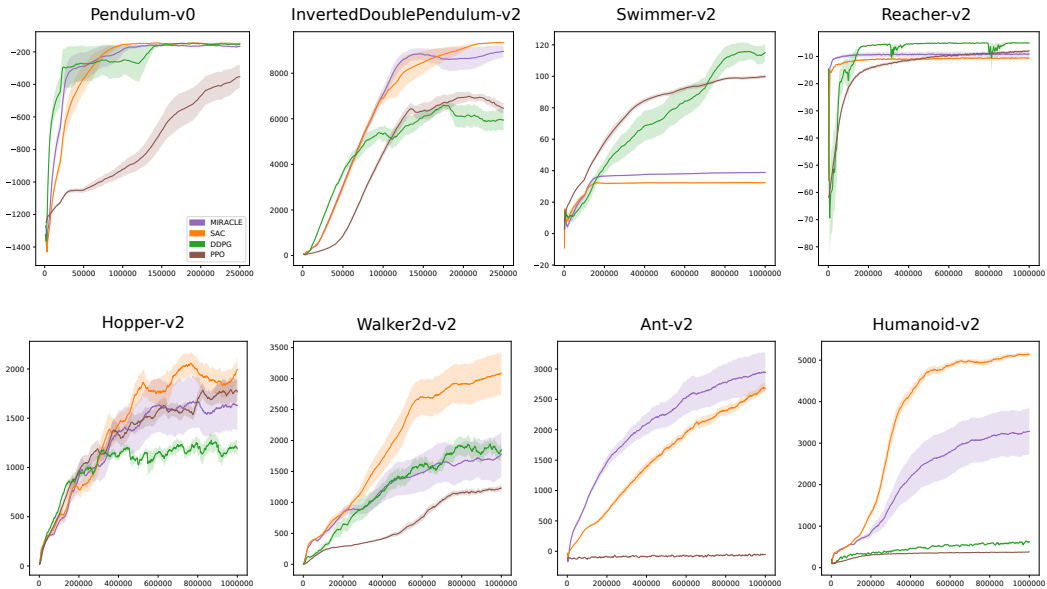


Figure 4: Mujoco Variational-Inference Experiments. The plot reports experiments with a variational autoencoder to model marginal actions. Results are depicted the same way as in Figure 2 from the main paper. Overall, results are worse compared to the main paper. The experimental setup was however exactly the same, i.e. both the generative and the recognition model performed a single parameter update for one minibatch only in each step. We hypothesize a better result could be obtained under a different training scheme that allows the autoencoder more training samples.

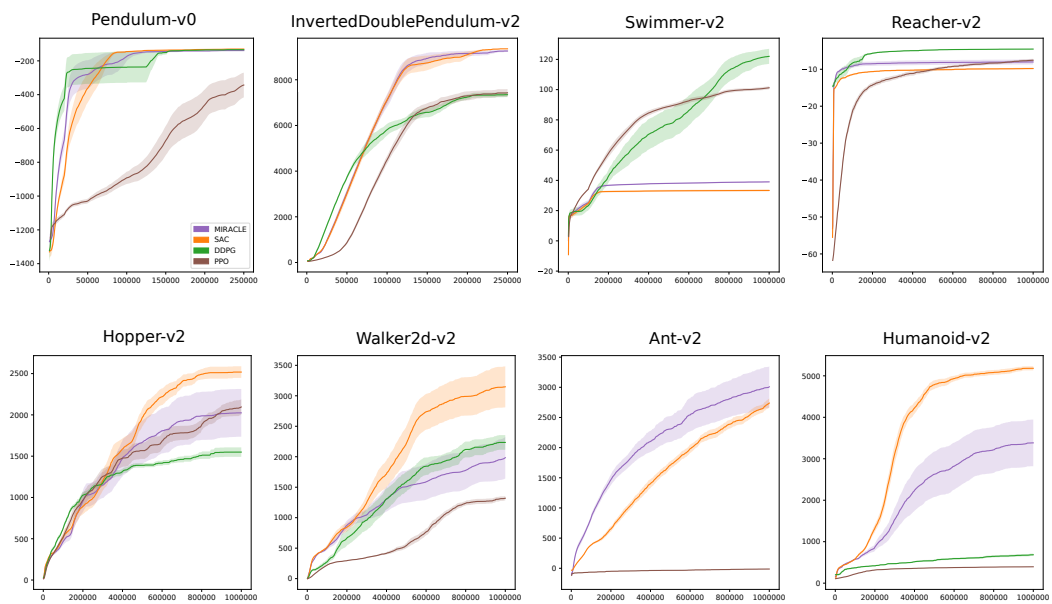


Figure 5: Mujoco Variational-Inference Experiments Best Rewards. Results are depicted similar to Figure 3 but for the variational autoencoder experiments.