Abstract

We develop a new measure of the exploration/exploitation trade-off in infinite-horizon reinforcement learning problems called the occupancy information ratio (OIR), which is comprised of a ratio between the infinite-horizon average cost of a policy and the entropy of its long-term state occupancy measure. The OIR ensures that no matter how many trajectories an RL agent traverses or how well it learns to minimize cost, it maintains a healthy skepticism about its environment, in that it defines an optimal policy which induces a high-entropy occupancy measure. Different from earlier information ratio notions, OIR is amenable to direct policy search over parameterized families, and exhibits hidden quasiconcavity through invocation of the perspective transformation. This feature ensures that under appropriate policy parameterizations, the OIR optimization problem has no spurious stationary points, despite the overall problem’s nonconvexity. We develop for the first time policy gradient and actor-critic algorithms for OIR optimization based upon a new entropy gradient theorem, and establish both asymptotic and non-asymptotic convergence results with global optimality guarantees. In experiments, these methodologies outperform several deep RL baselines in problems with sparse rewards, where many trajectories may be uninformative and skepticism about the environment is crucial to success.

1 Introduction

A critical, recurring dilemma in reinforcement learning (RL) and related sequential decision-making problems can be framed as follows: how should an agent balance the need to acquire information about its environment with the desire to exploit the resulting knowledge to achieve its goals? The field of RL \cite{Sutton2018} has seen many attempts to address this so-called exploration/exploitation trade-off by incentivizing exploration alone; the hope is that, with improved knowledge about an uncertain environment, the agent can eventually improve its exploitation capabilities. In control and autonomy, this dilemma is often addressed by separating the steps of planning and control: first one conducts state estimation (system identification \cite{Ljung1999}) over a stochastic system, and then solves for actuation parameters in terms of it \cite{Lee2002}. Trying to address these issues jointly has led many modern RL methods to include some augmentation to encourage exploration \cite{Sutton2018, Konda2000, Mnih2013, Silver2016}. These approaches define an intrinsic reward associated with exploration, rather than the extrinsic rewards traditionally associated with a Markov decision process (MDP), based off the tacit acknowledgement that overly honing in on an extrinsic reward may lead to narrow and occasionally spurious behavior. Very few works have directly quantified and optimized the exploration/exploitation trade-off, however, which leads us to pose the following question:

Can we precisely balance how much reward a policy seeks to accumulate during training with the inherent skepticism demanded by an uncertain environment?

Prior works in multi-armed bandits and RL \cite{Garivier2011, Agarwal2016} seek to balance the goals of exploration and exploitation by minimizing the ratio of cost incurred – formulated as regret – to information acquired when
determining which action or sequence of actions to choose. See Section 3 for an overview of this line of work. One of the key insights of these papers is that, though the ultimate goal of the agent may be to accumulate maximum reward, the randomness that connects rewards and actions demands addressing an auxiliary goal of acquiring more information about the environment by choosing informative actions. Put another way, increasing confidence based on experience is only beneficial when that experience is noise-free. Otherwise, action selection must retain a notion of respect for the unknown, as quantified by action informativeness. This fact is well-established in the multi-armed bandit literature, dating back to at least [43, 45, 24].

Our goal is to develop information ratio optimization approaches for infinite-horizon RL problems that can operate in high-dimensional, possibly continuous spaces. Prior works [33, 28] define an information ratio in terms of information gain to a time-horizon in the future, which in the infinite-horizon setting requires conditioning over an infinite trajectory. Moreover, tabular value functions are employed, which exhibit scaling proportional to the cardinality of the state and action spaces. To enable operation in high-dimensional, possibly continuous spaces, operating in parameter space rather than tabular representations is required, for which policy gradient methods are most natural [27, 37, 17]. In addition, recent theoretical progress has been made in providing global optimality guarantees for policy gradient methods even in the parameterized setting, under suitable conditions [9, 2, 29, 48, 7].

Our goal then demands defining a notion of informativeness that is amenable to policy search in parameter space. To do so, it is advantageous to seek informative policies instead of searching over a potentially large or infinite action space for informative actions. This raises a crucial question: how should the informativeness of policies be quantified? Occupancy measure entropy has recently been used as an optimization objective that quantifies the expected amount of exploration of the state (or state-action) space that a policy performs [18, 26, 48]. In other words, occupancy measure entropy quantifies the amount of information about the environment that a policy provides, on average, by measuring how uniformly the policy covers the state (or state-action) space. In this paper, we take the occupancy measure entropy, or occupancy information, of a policy to be the fundamental quantity defining its informativeness.

With this definition, then, we develop policy gradient and actor-critic algorithms that optimize a notion of the exploration/exploitation trade-off encapsulated by the ratio of long-term average cost to information acquired as quantified by occupancy measure entropy. We call this new objective the occupancy information ratio, or OIR; see Sections 3 and 4 for a granular discussion of how this ratio contrasts with other related definitions and information-theoretic quantities in RL. The OIR methods that we develop have several attractive properties. First, they provide a concise measure of the exploration/exploitation trade-off of policies that is simultaneously a tractable objective function. Second, due to the simplicity and tractability of the OIR, the algorithms are straightforward to understand and implement. Third, they have a rich underlying theory with strong convergence guarantees. Finally, in highly uncertain or sparse-reward environments where a healthy skepticism is essential to success, the OIR yields policies that avoid spurious, suboptimal behavior. In this paper, we establish and explore these properties with the following contributions.

- We propose a new RL objective, the occupancy information ratio (OIR), defined as the ratio of long-run average cost to occupancy information of a policy. With its natural interpretation as cost per unit information acquired about the environment, this objective provides a useful new measure of the exploration/exploitation trade-off in infinite-horizon reinforcement learning problems.
- Drawing on connections between the OIR optimization problem, quasiconcave optimization, and the powerful linear programming theory for Markov decision processes, we derive a concave programming reformulation of the OIR optimization problem over the space of state-action occupancy measures. This establishes a rich underlying theory that we exploit to strengthen our subsequent convergence results.
- We prove an entropy gradient theorem allowing us to perform policy gradient ascent/descent on objectives containing the occupancy measure entropy. This result is key to proving our subsequent OIR policy gradient theorem, but it is also of independent interest. We then leverage the OIR policy gradient theorem to develop two policy gradient algorithms for optimizing the OIR: Information-Directed REINFORCE (ID-REINFORCE) and Information-Directed Actor-Critic (IDAC).
- We establish the convergence theory underpinning OIR policy gradient methods with three key results: (1) due to the aforementioned concave programming reformulation, the OIR optimization problem enjoys a powerful hidden quasiconcavity property guaranteeing that its stationary points are in fact
global optima; (2) the fundamental gradient descent scheme underlying ID-REINFORCE enjoys a nonasymptotic convergence rate that depends in an interesting way on the parametrized policy class and other factors; (3) IDAC converges with probability one to (a neighborhood of) a global optimum of the OIR problem.

- We provide experiments illustrating the performance and theory of the IDAC algorithm. The first group of experiments demonstrates that IDAC can lead to improved performance in sparse-reward settings, even outperforming state-of-the-art deep RL algorithms like Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Q-learning (DQN) in certain cases. This provides empirical evidence supporting that the OIR yields policies avoiding spurious behavior. The second set of experiments illustrates the correctness of our theoretical results by showing that IDAC converges to the optimal solution predicted by the concave programming reformulation of the OIR optimization problem.

The remainder of the paper is organized as follows. First, in Section 2 we formally define the OIR objective and corresponding optimization problem. Next, in Section 3 we provide an overview of the related literature. Third, in Section 4 we compare and contrast existing information ratios with the OIR, then discuss why the OIR is well-suited to infinite-horizon policy search. Our main results begin in Section 5 where we first develop the concave programming reformulation of the OIR optimization problem, then derive the entropy gradient and OIR policy gradient theorems. In Section 6 we develop the ID-REINFORCE and IDAC algorithms. The remainder of our theoretical contributions are contained in Section 7 which presents our hidden quasiconcavity and convergence results. Finally, our experimental results are provided in Section 8, after which we conclude and discuss important future directions.

2 Problem Formulation

In this section, we develop the occupancy information ratio (OIR) objective. As we will show, the OIR can be optimized using policy gradient methods, has no spurious local extrema under suitable conditions, and algorithms optimizing it exhibit robust theoretical convergence guarantees. As discussed in Section 3, this state of affairs contrasts with existing information ratio formulations, which, though they provide powerful tools for theoretical analysis, are often difficult to optimize directly.

2.1 Occupancy Information Ratio

We first define an underlying Markov decision process, then formulate the OIR as an objective to be optimized over it.

Markov Decision Processes. Consider an average-cost Markov decision process (MDP) described by the tuple \((\mathcal{S}, \mathcal{A}, p, c)\), where \(\mathcal{S}\) is the finite state space, \(\mathcal{A}\) is the finite action space, \(p : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{D}(\mathcal{S})\) is the transition probability kernel mapping state-action pairs to distributions over the state space, and \(c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^+\) is the cost function mapping state-action pairs to positive scalars. In this discrete-time sequential decision-making setting, at a given timestep \(t\), the agent is in state \(s_t\), chooses an action \(a_t\) according to some policy \(\pi : \mathcal{S} \rightarrow \mathcal{D}(\mathcal{A})\) mapping states to distributions over the action set, incurs a corresponding cost \(c(s_t, a_t)\), and then the system transitions into a new state \(s_{t+1} \sim p(\cdot|s_t, a_t)\). Since we are primarily interested in policy gradient methods, we give the following definitions with respect to a parameterized family \(\{\pi_\theta : \mathcal{S} \rightarrow \mathcal{D}(\mathcal{A})\}_{\theta \in \Theta}\) of policies, where \(\Theta\) is some set of permissible policy parameters. Note that analogous definitions apply to any policy \(\pi\). For any \(\theta \in \Theta\), let

\[
\lambda_\theta(s, a) = \lim_{t \to \infty} P(s_t = s, a_t = a | \pi_\theta)
\]

denote the steady-state occupancy measure over \(\mathcal{S}\) induced by \(\pi_\theta\), which we assume to be independent of the initial start-state\(^1\). In addition, let

\[
\lambda_\theta(s, a) = \lim_{t \to \infty} P(s_t = s, a_t = a | \pi_\theta)
\]

\(^1\)Note that different from the discounted case, this expression is defined directly in terms of a limiting probability, rather than a sum over discounted state transitions.
Finally, let
\[
J(θ) = \lim_{n→∞} \frac{1}{n} \mathbb{E}_{π_θ} \left[ \sum_{i=1}^{n} c(s_i, a_i) \right] = \sum_s d_θ(s) \sum_a π_θ(a|s)c(s, a)
\]  

(3)

denote the long-run average cost of using policy π_θ.

**Occupancy Measure Entropy.** Given θ, define the entropy of the state occupancy measure induced by π_θ to be
\[
H(λ_θ) = -\sum_s \sum_a λ_θ(s, a) \log λ_θ(s, a) = -\sum_s d_θ(s) \sum_a π_θ(a|s) \left[ \log d_θ(s) + \log π_θ(a|s) \right].
\]  

(5)

Though we focus primarily on the state occupancy measure entropy (4) in this paper, the state-action occupancy measure entropy (5) will be considered in Corollary 1 to Theorem 2.

**Occupancy Information Ratio.** In this paper we consider the OIR objective
\[
ρ(θ) = \frac{J(θ)}{κ + H(d_θ)},
\]  

(6)

where κ ≥ 0 is a user-specific constant, discussed in Remark 1. When π_θ and the system dynamics are such H(d_θ) = 0 is possible, restricting κ > 0 ensures that (6) is well-defined. Under suitable conditions, however, H(d_θ) > 0 is guaranteed, so κ = 0 is allowed. Given an MDP \((S, A, p, c)\), our goal is to find a policy parameter θ∗ such that π_θ∗ minimizes (6), subject to the costs c and transition dynamics p of the MDP. For brevity, we say the goal is to minimize ρ(θ) over \((S, A, p, c)\). Note that the average cost J(θ) and entropy H(d_θ) are both long-run, infinite-horizon quantities, since their values depend on the asymptotic behavior of policy π_θ. For this reason, we regard the objective ρ(θ) as an infinite-horizon objective.

**Remark 1.** Since it scales the relative importance of the entropy term in the ratio, the constant κ plays the role of a regularizer in the denominator. When minimizing a function f(x), one frequently adds regularization to obtain a regularized objective function: f(x) + κ ∥x∥, where κ ≥ 0. In this setting, the larger κ becomes, the more important the regularization term becomes with respect to the objective function. In contrast, for the OIR defined in equation (6), the relative importance of the entropy term actually diminishes as κ becomes larger: when κ is small, even minor changes in the value of H(d_θ) can have a large effect on the value of ρ(θ); when κ is large, on the other hand, even significant perturbations of the value of H(d_θ) will have little effect on the value of ρ(θ).

### 3 Related Work

A large number of previous works have incorporated information-theoretic tools and related exploration-encouraging methods into sequential decision-making problems in an effort to tackle the exploration/exploitation trade-off. In this section we begin with a literature review of related efforts for the multi-armed bandit (MAB) and reinforcement learning (RL) settings.

\[
\text{Since one typically speaks of entropy of a random variable, not entropy of an occupancy measure, this definition of } H(d_θ) \text{ is a slight abuse of notation. Strictly speaking, we are using } H(d_θ) \text{ as shorthand for } H(S) = -\sum_s d_θ(s) \log d_θ(s), \text{ where } S \text{ is an } S\text{-valued random variable with probability mass function } d_θ(\cdot).
\]

\[
\text{One common situation where } H(d_θ) > 0 \text{ holds, for all } θ ∈ Θ, \text{ is when all Markov chains induced by the members of the policy class } \{π_θ\}_{θ∈Θ} \text{ over } S \text{ are assumed to be ergodic.}
\]
Multi-Armed Bandits. In the MAB setting, classic approaches include Gittins index methods, upper confidence bound (UCB) methods, Thompson sampling (TS), and information-directed sampling (IDS). Viewing the MAB as a certain class of optimal stopping problem, it was first shown in [15] how to obtain an optimal policy via computation of the Gittins index. Though Gittins index-based methods are theoretically guaranteed to provide optimal policies, computational methods based on them, such as that outlined in [25, §35.5], rely on knowledge or a learned model of the system dynamics, making them impractical for complex problems. UCB methods [1, 13, 38] use an optimistic over-estimate of action rewards coupled with greedy policies to encourage exploration of infrequently chosen actions. TS schemes [3], on the other hand, iteratively update the agent’s belief about the identity of the optimal action based on experience, all while choosing actions as if the current belief is optimal. Interestingly for our setting, [34] uses an information ratio for MABs, defined in equation (7) below, as an analysis tool to prove that TS leads to superior regret bounds. A work related to [34, 33] develops the IDS scheme to balance regret with information acquisition by explicitly minimizing the MAB information ratio at each timestep, leading to better bounds than both TS and UCB in many cases. However, it is often unclear how to use the information ratio in MAB settings to devise new algorithms.

Information in Reinforcement Learning. Moving to the MDP setting, a vast literature has grown up around methods for balancing exploration and exploitation. Within this literature, many works leverage information-theoretic connections to encourage exploration of the state and action spaces. An early connection between policy gradient methods and information-theoretic techniques is the introduction of Fischer information-based preconditioners for gradient updates in the natural policy gradient method [21, 20], which improves gradient updates by taking the information geometry of the policy parameterization into account. More recently, policy entropy regularization, where a bonus is provided for policies that behave more randomly, has been shown to improve practical success [17] and leads to policies that inherit favorable theoretical attributes of proximal point methods [29]. Similar to the MAB setting, UCB-type exploration bonuses have also been successfully used to encourage exploration in the RL setting. In particular, certain forms of UCB-induced exploration lead to regret bounds that are comparable to model-based methods when used in conjunction with the model-free Q-learning algorithm in the tabular case [19]. In a distinct, but related direction, the recent work [14] proposes a novel, information-theoretic measure of task complexity, called policy-(optimal) information capacity, and empirically demonstrates how it can be used to determine the difficulty of a range of common RL problems.

Maximum Entropy Exploration. In recent years RL for maximum entropy exploration has seen increasing interest. Unlike the policy entropy regularization described above, the schemes developed in [18, 48] provide a way to directly maximize the entropy $H(d_\theta)$ of the state occupancy measure $d_\theta$. In the absence of a reward signal, [18] argue that it is reasonable for an agent to instead maximize some measure of coverage of the state space, such as $H(d_\theta)$. The authors develop a novel algorithm for maximizing a concave reward functional of the state occupancy measure (which includes entropy), and, given access to suitable oracles, they prove convergence to a nearly optimal policy and provide sample complexity bounds. Similar to [18], the variational policy gradient algorithm proposed in [18] studies the problem of optimizing reward functionals that are not as nicely behaved as the standard discounted reward. The saddle-point algorithm that they develop can be applied to maximize any objective that can be formulated as a concave function of the state-action occupancy measure. Furthermore, it is shown by introducing the notion of hidden concavity that, when viewed as an optimization problem over the parameter space, the problem has no spurious local extrema, i.e., every stationary point is global optimal. This notion of hidden concavity is important for our current work, as we generalize it to prove a similar result for our algorithms. Though it does not explicitly consider entropy maximization, the recent work [47] explores a new class of risk-sensitive RL methods by considering optimization over the space of state-action occupancy measures. This is closely related to both [35] and our current paper, since in all three the dual of the primal linear programming formulation of the underlying MDP is modified to incorporate new learning objectives: exploration in [35] and the present work, and risk-sensitivity in [47].

---

4Policy entropy is defined as $H(\pi(\cdot|s)) = -\sum_a \pi(a|s) \log \pi(a|s)$. When used as a regularizer, larger values of $H(\pi(\cdot|s))$ are encouraged.
Information Ratios. Of all the related works described in this section, the work \cite{28} on IDS for MABs discussed above and the recent work \cite{33} developing an information ratio for general problems are most relevant to our current paper. In \cite{28}, the authors develop a powerful, abstract framework for reasoning about the design of algorithms for solving sequential decision-making problems, including MDPs. They generalize the MAB information ratio of \cite{33} to this setting, resulting in the \( \tau \)-information ratio defined in equation (8) below, and regret bounds and algorithmic schemes are provided. In the IDS scheme proposed in \cite{33}, the information ratio is minimized at each step to obtain an action-selection distribution that balances minimizing regret with gaining information about the optimal action. One of the goals of \cite{28} is to use the \( \tau \)-information ratio to develop similar sampling schemes for the RL setting to guide action selection at each step. Due to inherent difficulties in minimizing regret and estimating mutual information, the authors acknowledge that optimizing their information ratio directly is typically intractable. For this reason they develop a series of useful surrogate or \textit{pseudo}-information ratios to optimize instead, resulting in the value-IDS and variance-IDS schemes. Variance-IDS, the more computationally tractable of the two, requires maintaining and updating belief distributions over the set of possible action-value functions as well as solving a convex optimization problem over the action set at each step. This restricts its applicability to smaller problems and makes it difficult to scale to continuous action spaces. The use of epistemic neural networks to overcome the problem of maintaining belief distributions is explored, however, and promising empirical results are provided demonstrating performance comparable to or better than TS on problems where information acquisition is key.

4 Broadening the Information Ratio’s Horizons

Our goal in this paper is to develop policy search methods for optimizing the rate of exploitation per unit exploration in the infinite-horizon RL setting that are scalable to large state and action spaces through suitable use of policy and value function parameterization. We propose to do this by minimizing the OIR defined in (6). As discussed in the related works section, however, information ratios that capture a certain notion of exploitation per unit exploration have already been proposed. It is therefore natural to ask why a new type of information ratio is necessary; cannot policy search methods be developed for optimizing existing information ratios? To answer this question, in this section we review existing information ratio definitions from the multi-armed bandit and recent RL literatures. We then examine the difficulties that arise when trying to formulate policy search schemes for these information ratios. Finally, we explain why the OIR objective is better-suited for policy search in infinite-horizon RL, and how it exhibits better scaling with respect to the cardinality of the state and action spaces.

4.1 Existing Information Ratios

We now present the information ratios developed in the previous works \cite{33} and \cite{28}. This will allow us to highlight the key differences from the OIR in the following section.

Multi-Armed Bandits. We first discuss the information ratio considered in \cite{33} \cite{34}. Let a finite set of actions \( A \), a set \( \mathcal{Y} \) of outcomes, and a reward function \( R : \mathcal{Y} \to \mathbb{R} \) be given. In the multi-armed bandit (MAB) problem, at timestep \( t \in \mathbb{N} \), the agent chooses action \( a_t \in A \), observes outcome \( Y_{t,a} \in \mathcal{Y} \), and receives reward \( R_{t,a} = R(Y_{t,a}) \). Let \( Y_t = [Y_{t,a}]_{a \in A} \) denote the vector of possible outcomes at time \( t \), and assume there exists a random variable \( Z \) such that the sequence of random variables \( \{Y_t \mid Z\}_{t \in \mathbb{N}} \) conditioned on \( Z \) is i.i.d. Given \( Z \), denote the optimal (or an optimal, in the case of multiple optima) action by \( A^* \in \arg \max_a E[R_{1,a} \mid Z] \).

Let \( \mathcal{F}_t = \sigma(A_1, Y_1, A_1, \ldots, A_{t-1}, Y_{t-1}, A_{t-1}) \) denote the \( \sigma \)-algebra generated by the trajectory up to time \( t \). Define a policy \( \pi = \{\pi_t\}_{t \in \mathbb{N}} \) as a sequence of mappings \( \pi_t : \mathcal{F}_t \to \mathcal{D}(A) \) from trajectories to distributions over the action set. Furthermore, let \( \alpha_t(a) = P(A^* = a \mid \mathcal{F}_t) \) denote the posterior distribution of \( A^* \) given \( \mathcal{F}_t \). Define the instantaneous expected regret of choosing a given action at timestep \( t \) to be

\[
\Delta_t(a) = E [R_{t,A^*} - R_{t,a} \mid \mathcal{F}_t].
\]
Given two discrete random variables $X$ and $Y$ defined on the same probability space, the mutual information $I(X; Y)$ between $X$ and $Y$ is defined in [1] to be

$$I(X; Y) = D_{KL} [P_{X,Y} \parallel P_X \times P_Y] = \sum_u \log \left( \frac{P_{X,Y}(u)}{P_X(u)P_Y(u)} \right) P_{X,Y}(u),$$

the Kullback-Leibler divergence between the joint distribution of $X, Y$ and the product of their marginals. The mutual information has the following property:

$$I(X; Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X),$$

where $H(X) = -\sum_x \log(p_X(x))p_X(x)$ and $H(X \mid Y) = -\sum_y p_Y(y)\sum_x \log(p_{X\mid Y}(x \mid y))p_{X\mid Y}(x \mid y).$ The conditional mutual information of random variables $X, Y$ conditioned on $\mathcal{F}_t$, called the mutual information under the posterior distribution in [33], is given by

$$I_t(X; Y) = D_{KL} [P((X, Y) \in \cdot \mid \mathcal{F}_t) \parallel P(X \in \cdot \mid \mathcal{F}_t) \times P(Y \in \cdot \mid \mathcal{F}_t)]$$

The information gain at time $t$ of action $a$ is defined in [33] to be $g_t(a) = I_t(A^*; Y_{t,a})$ and satisfies the equality

$$g_t(a) = I_t(A^*; Y_{t,a}) = \mathbb{E} \left[ H(\alpha_t) - H(\alpha_{t+1}) \mid \mathcal{F}_t \right],$$

which can be understood as the reduction in uncertainty in the posterior over the optimal action. Finally, given an action sampling distribution $\pi \in \mathcal{D}(A)$, the expected instantaneous regret and information gain are given by

$$\Delta_t(\pi) = \sum_a \pi(a)\Delta_t(a)$$

and

$$g_t(\pi) = \sum_a \pi(a)g_t(a),$$

respectively.

**Information Ratio for MABs.** At time $t$, then, the information ratio of an action sampling distribution $\pi$ is defined as [33]

$$\Psi_t(\pi) = \frac{[\Delta_t(\pi)]^2}{g_t(\pi)},$$

providing a useful quantity relating the squared instantaneous regret to the information gained about the optimal action by following policy $\pi$. In addition to its use as an analysis tool for obtaining improved regret bounds [34], the information ratio (7) is used to guide exploration using the information-directed sampling scheme presented in [33] for optimizing bandit problems. This amounts to finding and sampling according to a $\pi$ minimizing (7) at each timestep. This objective balances minimizing regret in the numerator with maximizing the information gained about the optimal action in the denominator. Note that, unlike the infinite-horizon objective [3], the numerator and denominator in (7) are both finite-horizon, single-timestep quantities.

**General Sequential Decision-Making Problems.** We now turn to the general information ratio defined in [28]. Consider an environment given by $\mathcal{E} = (\mathcal{A}, \mathcal{O}, p)$, where $\mathcal{A}$ is a finite action set, $\mathcal{O}$ is a finite observation set, and the transition function $p : \mathcal{H} \times \mathcal{A} \rightarrow \mathcal{D}(O)$ gives a probability distribution $p(\cdot \mid h, a)$ over $\mathcal{O}$ for each history-action pair $(h, a) \in \mathcal{H} \times \mathcal{A}$, where $\mathcal{H}$ is the set of all possible state-observation trajectories, i.e. $\mathcal{H} = \{(o_0, o_1, o_2, \ldots, o_t) \mid t \in \mathbb{N}\}$. Furthermore, let a reward function $r : \mathcal{H} \times \mathcal{A} \times \mathcal{O} \rightarrow \mathbb{R}$ be given. In this general setting, at timestep $t \in \{0, \ldots, T-1\}$, for some integer $T$, and equipped with policy $\pi : \mathcal{H} \rightarrow \mathcal{D}(A)$, the agent observes the current history $h_t$, selects an action $a_t \sim \pi(\cdot \mid h_t)$, sees observation $o_{t+1}$, receives reward $r_t = r(h_t, a_t, o_{t+1})$, and finally sets $t \leftarrow t + 1$ and repeats until $t = T$. Note that this general formulation contains MDPs as a special case.

Given a policy $\pi$ and history $h$, let $V_\pi(h) = \mathbb{E}_\pi \left[ \sum_{t=|h|}^{T-1} r_t \mid h \right]$ denote the expected total reward of following policy $\pi$ starting from history $h$ onward, where $|h|$ denotes the time index of the last element of the trajectory $h$. Similarly, let $Q_\pi(h, a) = \mathbb{E}_\pi \left[ \sum_{t=|h|}^{T-1} r_t \mid h, a \right]$ denote the expected total reward of taking action $a$ at history $h$, then afterwards following policy $\pi$. Furthermore, let $V_\pi(h) = \sup_a Q_\pi(h, a)$ and $Q_\pi(h, a) = \sup_a Q_\pi(h, a)$ denote the optimal value and action-value functions, respectively. Using these, the expected immediate shortfall is defined as $\mathbb{E}_\pi [V_\pi(h_t) - Q_\pi(h_t, a_t)]$. For the precise definitions of all of these quantities, see [28].
Next, let the random variable $\chi$ denote the abstract learning target of the agent, representing its goals while exploring the environment $\mathcal{E}$. Furthermore, let $P_t$ denote the agent’s learned, finite representation of $\mathcal{E}$ at time $t$, called the epistemic state of the agent. This can include parameters of any learned models of the environment, such as value function approximators or policies. Notice that, conditional on $P_t$, $\mathcal{E}$ can be viewed as a random variable from the agent’s perspective. Recalling the definition of conditional mutual information from above, $\mathbb{I}(\chi; \mathcal{E}|P_t)$ can be understood as the mutual information between the learning target $\chi$ and environment $\mathcal{E}$ given the agent’s current epistemic state $P_t$. Given a number of timesteps $\tau$, the quantity $\mathbb{I}(\chi; \mathcal{E}|P_t) - \mathbb{I}(\chi; \mathcal{E}|P_{t+\tau})$ thus represents the total reduction in uncertainty about the learning target between timesteps $t$ and $t + \tau$. Again, for precise definitions and a thorough discussion, we refer the reader to [28].

**General $\tau$-Information Ratio.** With the foregoing in mind, the $\tau$-information ratio from time $t$ to time $t + \tau$ is defined as

$$
\Gamma_{\tau,t} = \frac{\mathbb{E}[V_*(h_t) - Q_*(h_t, a_t)]^2}{\mathbb{E}(\chi; \mathcal{E}|P_t) - \mathbb{E}(\chi; \mathcal{E}|P_{t+\tau})}/\tau,
$$

where the expectation in the numerator is taken with respect to the agent’s policy at time $t$. When starting with a policy $\pi$ at time $t$, this quantity has an intuitive interpretation as the ratio of the squared immediate shortfall to the average information gained about the learning target $\chi$ over the subsequent $\tau$ timesteps. Similar to (7) for the MAB setting, the $\tau$-information ratio is useful as a theoretical tool for obtaining regret bounds. Though the authors recognize that directly optimizing (5) is difficult in general, they propose optimizing various proxies, ultimately proposing variants of Q-learning with exploration strategies based on minimizing the $\tau$-information ratio. However, without any parameterization of the the reduction in mutual information between the epistemic state and the learning target of the agent, optimizing this quantity is intractable for large spaces. Moreover, like its predecessor (7), and unlike (6), the ratio (8) is a decidedly finite-horizon objective. We will have more to say about this in the following section.

### 4.2 Why the Occupancy Information Ratio?

We next outline the difficulties in applying policy search to the information ratio (5) developed in [28] and discuss why our OIR objective (6) is better-suited for policy search algorithms in the infinite-horizon RL setting.

In this paper we focus on developing policy search methods that optimize exploitation per unit exploration. We focus in particular on the infinite-horizon setting, which is the de facto standard in the policy gradient literature and enjoys well-developed underlying theory. Though we consider finite $\mathcal{S}$ and $\mathcal{A}$ for the purposes of analysis, an additional goal of this work is to lay the foundations for future development of practical methods that scale to continuous spaces; the success of policy search methods in such settings [36] motivates our focus on policy search. At the same time, since we are interested in optimizing exploitation per unit exploration, we naturally look to the information ratio work [33] and especially [28] for inspiration. However, attempting to perform infinite-horizon policy search using the information ratio (8) proposed in [28] directly is not practicable for two main reasons: first, the $\tau$-information ratio (5) is fundamentally tied to the finite-horizon setting; second, the proxies for (8) suggested in [28] are only tractable in the finite action space case. This motivates our development of the OIR as a surrogate for optimizing exploitation per unit exploration, since it is well-suited to policy gradient methods and leads to algorithms with robust convergence theory.

To see why the $\tau$-information ratio (5) is fundamentally tied to the finite-horizon setting, note that the value and action-value functions $V_*$ and $Q_*$ in the numerator are defined with respect to a fixed horizon $T$, while the epistemic states $P_t$ and $P_{t+\tau}$ are separated by a fixed, finite interval $\tau$. This dependence on $T$ makes development of scalable policy search difficult, as existing policy gradient techniques are overwhelmingly for the infinite-horizon case. Even if dependence of the numerator of (8) on $T$ can be removed, the problem still remains of how to deal with the finite $\tau$ in the denominator. A simple idea is to take $\tau \to \infty$ and consider $\lim_{\tau \to \infty} \mathbb{I}(\chi; \mathcal{E}|P_{t+\tau})$, but this leaves us with two problems: first, how do we interpret conditioning mutual information on an infinite trajectory? Second, given the difficulty of estimating mutual information conditioned on a finite trajectory, how can we estimate it when conditioned on an infinite one? It is unclear whether these questions have satisfactory answers.
Nonetheless, while developing the value-IDS and variance-IDS schemes [cf. Sec. 3], [28] proposes several proxy ratios for (8). Though these proxies are important and useful guides for short-term exploration, all are still firmly planted in the finite-horizon setting, without clear infinite-horizon analogues. Furthermore, assuming the issues of numerically estimating conditional mutual information and other hard-to-approximate quantities can be resolved, the use of these proxies still involves minimizing over the space $D(A)$ of probability distributions over the action space. This intrinsically links these proxy versions of (8) to the finite action space case, which runs counter to our goal of developing an information ratio-inspired policy search scheme that can eventually scale to continuous spaces.

For these reasons, we propose the OIR defined in (6) as a surrogate for optimizing exploitation per unit exploration via policy search in the infinite-horizon setting. The OIR enjoys several attractive qualities. First, it has a natural interpretation as expected cost per unit information gained about the environment, providing an intuitive measure of the rate of exploitation per unit exploration of a given policy. Second, due to the entropy gradient theorem proved below, the gradient of (6) can be easily estimated, enabling development of variants of classic policy gradient schemes, such as REINFORCE [46] and actor-critic [21,10], in parameterized settings that exhibit tractable scaling to large spaces. In particular, due to the ease with which our framework admits the use of neural network function approximators, OIR variants of modern deep learning-based policy optimization schemes [36,37] are possible. Finally, methods based on the OIR enjoy robust underlying theory and strong convergence guarantees, as well as promising empirical performance.

5 Elements of Occupancy Information Ratio Optimization

We now turn to the problem of optimizing the occupancy information ratio (OIR) defined in (6). In this section we develop two key aspects of the theory underlying this problem.

First, we build on parallels with linear programming solutions to Markov decision processes (MDPs) to show that, given access to the cost function and transition dynamics of the underlying MDP, we can transform the non-convex problem of minimizing our objective (6) into a concave program over the space of state-action occupancy measures. This transformation relies in a crucial way on our definition of the OIR, which ensures that (6) is quasiconvex in the state-action occupancy measure and unlocks the concave reformulation. In addition to laying firm theoretical foundations for OIR minimization by drawing connections to the rich linear programming theory for MDPs, this concave reformulation allows us to efficiently recover optimal policies in certain cases, which we use to provide optimal benchmarks in our experimental results. Furthermore, as discussed in Section 7.1, the existence of this concave reformulation endows the OIR optimization problem with a powerful hidden quasiconcavity property that we exploit to strengthen the convergence results for our policy gradient algorithm.

Second, since the cost and transition dynamics are typically unknown in practice, in the second part of this section we lay the groundwork for model-free policy search methods by deriving a policy gradient theorem providing access to the gradient of the OIR objective (6) over the space of state-action occupancy measures. This critical reliance on the definition of the OIR, as it admits a tractable gradient expression that is easy to estimate in practice. On the road to proving this result, we derive an entropy gradient theorem that provides a simple expression for the gradient of the entropy, $\nabla_{\theta} H(d_{\theta})$; we believe this entropy gradient result is of independent interest. Finally, the gradient expressions developed here are used in the following section to develop policy gradient algorithms for minimizing (6).

5.1 Concave Reformulation

In this subsection we leverage connections with the theory of linear programming for MDPs to show that, given knowledge of the cost function and transition probability function of an MDP, we can solve a related concave program to find a policy minimizing the OIR objective (6) over the MDP.

Given an average-cost MDP $(S, A, p, c)$ and a policy $\pi$, let $\lambda_{\pi} \in D(S \times A)$ denote the state-action occupancy measure induced by $\pi$ on $S \times A$, i.e. $\lambda_{sa} = \lim_{t \to \infty} P(s_t = s, a_t = a | \pi)$. As discussed in §8.8, if we have access to $p$ and $c$, an optimal state value function can be obtained by solving a related linear program, (LP). This is useful, as linear programs can be solved efficiently. Furthermore, the state-action occupancy measure $\lambda^*$ of the optimal policy $\pi^*$ for $(S, A, p, c)$ can be obtained by solving the following linear
The constraints ensure that the decision variables $\lambda$ give a valid state-action occupancy measure for the MDP, i.e. that the probability vector $\lambda \in \mathcal{D}(S \times A)$ is achievable given the transition probability function $p$. With this in mind, for a given $\lambda$, the objective function is clearly the expected long-run average cost of following a policy whose state-action occupancy measure is $\lambda$. Furthermore, given a feasible solution $\lambda$ to (D), the policy $\pi_\lambda$ defined by

$$
\pi_\lambda(a|s) = \frac{\lambda_{sa}}{\sum_{a'} \lambda_{sa'}}
$$

induces the state-action occupancy measure $\lambda$ (see [32, Thm. 8.8.2] for details). This means that, once the optimal $\lambda^*$ is obtained by solving (D), the corresponding policy $\pi_{\lambda^*}$ is optimal for $(S,A,p,c)$.

An analogous optimization problem can be formulated for minimizing $\rho$ over the MDP $(S,A,p,c)$. Consider the following:

$$
\min_{\lambda} \quad \rho(\lambda) = \frac{J(\lambda)}{\kappa + H(\lambda)} = \frac{c^T \lambda}{\kappa - \sum_s (\sum_a \lambda_{sa}) \log(\sum_a \lambda_{sa})}
$$

s.t. \quad \sum_s \sum_a \lambda_{sa} = 1,

$$
\sum_a \lambda_{sa} = \sum_s \sum_a p(s'|s,a) \lambda_{s'a}, \quad \forall s \in S,
$$

$$
\lambda \geq 0,
$$

where $\hat{H}(\lambda) = H(d^\lambda)$ is the entropy of the state occupancy measure $d^\lambda \in \mathcal{D}(S)$ given by $d^\lambda = \sum_a \lambda_{sa}$. Recall that in the standard definition of the function $H(d)$, for any $d_i = 0$, we take $d_i \log d_i = \lim_{d_i \to 0^+} d_i \log d_i = 0$, so $H(d)$ is always well-defined and finite for $d \geq 0$ (see, e.g., [16]). Similarly, we take $d^\lambda_i \log d^\lambda_i = 0$ whenever $d^\lambda_i = 0$, so that $\hat{H}(\lambda)$ is well-defined for $\lambda \geq 0$. To ensure that the objective $\rho(\lambda)$ is well-defined over the feasible region of (Q), we make the following mild assumption:

**Assumption 1.** Given any $\lambda$ feasible to (Q), the state occupancy measure $d^\lambda$ has at least two non-zero entries.

Assumption 1 amounts to stipulating that, for any policy $\pi$, the transition probability function $p$ is such that, in the long run, $\pi$ visits at least two states with strictly positive probability. In other words, $d_\pi = d^{\lambda_\pi}$ assigns positive probability to at least two distinct states. This guarantees that the denominator $\kappa + \hat{H}(\lambda)$ of $\rho(\lambda)$ is always strictly positive, even when $\kappa = 0$. When $d^\lambda$ has only one nonzero entry and $\kappa = 0$, on the other hand, $\rho(\lambda)$ will have zero denominator and thus be undefined. Note that Assumption 1 is much weaker than the condition that every policy induces an ergodic Markov chain on $S$, which is frequently encountered in the MDP and RL literature (see Assumption 2 below).

Since the feasible region of (Q) corresponds to precisely those state-action occupancy measures achievable over $(S,A,p,c)$, solving (Q) clearly yields the state-action occupancy measure minimizing the OIR $\rho(\lambda)$. Furthermore, as when solving the linear program (D) above, any optimal solution $\lambda^*$ to (Q) allows us to recover a policy $\pi_{\lambda^*}$ minimizing $\rho(\lambda)$. Unlike the linear program (D), however, the objective function in (Q) is non-convex, so the problem may be difficult to solve directly. Nonetheless, due to the quasiconvexity of $\rho(\lambda)$, we will show that problem (Q) can be transformed to an equivalent concave program. The latter can then be efficiently solved to obtain the optimal state-occupancy measure and corresponding optimal policy.
5.1.1 Quasiconvexity of (Q)

As a first step on our path to showing how to solve (Q) using a concave reformulation, we prove that the quasiconvex optimization problem (Q) is feasible and that every local optimal solution to it is a global optimal solution. We will then show that the problem can be equivalently formulated as an intermediate quasiconcave optimization problem. We subsequently show how this quasiconcave program can be transformed into a concave program via the range transformation described in [6, Ch. 7]. The latter transformation is our end goal, as the resulting concave program can then be efficiently solved to obtain an optimal solution to (Q).

Now that we are assured (Q) is feasible and has an optimal solution with finite objective function value, and the objective ρ of (Q) is strictly quasiconvex on F.

Proof. The problem is clearly feasible, since any arbitrary policy π induces a valid state-action occupancy measure λπ ∈ F. Since F is bounded as a subset of the unit simplex, and also J(λ) > 0 and κ + ̃H(λ) > 0 on F by Assumption [1] we have that ρ is bounded on F. Furthermore, since J(λ) is linear on F, κ + ̃H(λ) is concave on F, and J(λ) > 0 and κ + ̃H(λ) > 0 on all the sublevel sets of ρ on F, [6, Prop. 5.20] applies to show that ρ is quasiconvex on F.

Finally, (Q) enjoys the following key property, which guarantees that any solution to the concave program described in the next section provides a globally optimal solution to the OIR minimization problem (Q).

Lemma 2. Every local optimum of (Q) is a global optimum.

Proof. The assertion follows directly from [6, Prop. 3.3].

5.1.2 Transformation to a Concave Program

Now that we are assured (Q) is quasiconvex and has no spurious stationary points, we show it can be reformulated as an equivalent concave program. To do so, we first transform it into an equivalent quasiconcave maximization problem by showing that minimizing the objective function ρ(λ) of problem (Q) is equivalent to maximizing its reciprocal 1/ρ(λ). Next, we show that, by using a suitable change of variables first proposed in [35], this objective function can be reinterpreted as the perspective transform of a concave function, which is itself concave. Finally, we apply results from [6] to obtain the desired concave reformulation of (Q). As highlighted at the beginning of this section, the definition of the OIR is critical to this reformulation, as it exploits structural attributes of the family of state-action occupancy measures to enable the transformation from the initial quasiconvex program to the desired concave program.
Define \( q(\lambda) := 1/\rho(\lambda) = (\kappa + \hat{H}(\lambda))/J(\lambda) \) and consider the problem

\[
\begin{align*}
\max_{\lambda} & \quad q(\lambda) \\
\text{s.t.} & \quad \sum_{s} \sum_{a} \lambda_{sa} = 1, \\
& \quad \sum_{a} \lambda_{sa} = \sum_{s'} \sum_{a} p(s'|s, a) \lambda_{s'a}, \quad \forall s \in S, \\
& \quad \lambda \geq 0.
\end{align*}
\]

\( (Q_1) \)

Note that the feasible region \( F \) of \( (Q_1) \) is identical to the feasible region of \( (Q) \). We have the following result:

**Lemma 3.** Problem \( (Q_1) \) is equivalent to \( (Q) \).

**Proof.** Assume \( \lambda^* \) is optimal for \( (Q_1) \), i.e. \( \lambda^* \in F \) and \( \rho(\lambda^*) \leq \rho(\lambda) \), for all \( \lambda \in F \). By Lemma 1, there exists \( M > 0 \) such that \( 0 < \rho(\lambda^*) \leq 1 < M < \infty \), for all \( \lambda \in F \), so \( \lambda^* \) is optimal to \( (Q_1) \). By an analogous argument, any optimal solution to \( (Q_1) \) is optimal to \( (Q) \).

The foregoing Lemma proves that solving \( (Q_1) \) also solves \( (Q) \). Crucially, we can in fact transform \( (Q_1) \) into a *concave* optimization problem, which will allow us to indirectly solve \( (Q) \). Using the variable transformation \( y = \frac{\lambda}{c^T \lambda}, \quad t = \frac{1}{c^T \lambda} \), and applying the key result [6, Prop. 7.2], we obtain the equivalent problem

\[
\begin{align*}
\max_{y, t} & \quad t \left[ \kappa + \hat{H} \left( \frac{y}{t} \right) \right] \\
\text{s.t.} & \quad \sum_{s} \sum_{a} \frac{y_{sa}}{t^2} - 1 = 0, \\
& \quad \sum_{a} \frac{y_{sa}}{t^2} - \sum_{s'} \sum_{a} \frac{p(s'|s, a) y_{s'a}}{t^2} = 0, \quad \forall s \in S, \\
& \quad \sum_{s} \sum_{a} c_{sa} \frac{y_{sa}}{t^2} = 1, \\
& \quad y \geq 0, \quad \frac{y}{t} \geq 0.
\end{align*}
\]

\( (Q_2) \)

Notice that the constraint \( \sum_{s} \sum_{a} c_{sa} y_{sa} = 1 \), combined with the fact that \( c > 0 \), implies that \( y_{sa} > 0 \) for at least one \( y_{sa} \). Since \( y \geq 0 \) and \( \sum_{s} \sum_{a} y_{sa} = t \), this means \( t > 0 \). The constraint \( y/t \geq 0 \) is thus redundant, so it can be removed. The objective and the first two constraints in problem \( (Q_2) \) can also be simplified to obtain the following, cleaner form:

\[
\begin{align*}
\max_{y, t} & \quad \kappa t - \sum_{s} \sum_{a} y_{sa} \log \left( \frac{\sum_{a} y_{sa}}{t} \right) \\
\text{s.t.} & \quad \sum_{s} \sum_{a} y_{sa} = t, \\
& \quad \sum_{a} y_{sa} = \sum_{s'} \sum_{a} p(s'|s, a) y_{s'a}, \quad \forall s \in S, \\
& \quad \sum_{s} \sum_{a} c_{sa} y_{sa} = 1, \\
& \quad y \geq 0.
\end{align*}
\]

\( (Q_3) \)

The constraints clearly define a convex feasible region, and the objective function is also concave, as it is the *perspective* of the function \( \kappa + \hat{H}(\lambda) \), which is defined for the general setting as follows:
**Definition 2.** Given \( f : \mathbb{R}^n \to \mathbb{R} \), the perspective of \( f \) is the function \( P_f : \mathbb{R}^{n+1} \to \mathbb{R} \) given by \( P_f(x,t) = tf(x/t) \) with domain \( \text{dom}(P_f) = \{(x,t) \mid x/t \in \text{dom}(f), t > 0\} \).

As discussed in [12 §3.6.2], the perspective operation preserves concavity/convexity. In particular, if \( f \) is concave (resp. convex), then \( g \) is concave (resp. convex). Thus \( (Q_3) \) is a concave program. Finally, given an optimal solution \( (y^*,t^*) \) to \( (Q_3) \), we can recover an optimal solution \( \lambda^* = y^*/t^* \) to \( (Q) \), which by Lemma 3 is also optimal for \( (Q) \). As noted above, the policy \( \pi^*(a|s) = \lambda^*_sa/\sum_a \lambda^*_sa' \) thus minimizes the OIR over the MDP \( (S,A,p,c) \). We formalize the foregoing as the following result.

**Theorem 1.** Any optimal solution to the concave program \( (Q_3) \) is optimal for the OIR problem \( (Q) \).

The goal of this section has now been achieved: to solve the non-convex OIR optimization problem, we can efficiently solve the concave program \( (Q_3) \) and use the solution to recover an optimal policy for the OIR problem. In addition, this reformulation implies the existence of hidden quasiconcavity underlying any policy gradient methods developed for the OIR minimization problem, as shown in Section 7.1. As we show later on, the existence of hidden quasiconcavity enables us to prove much stronger convergence results for our policy gradient methods than would otherwise be possible.

### 5.2 Policy Gradients

As shown above, we can efficiently solve the OIR optimization problem given knowledge of the cost and transition functions. In most cases the environment is unknown, however, so we resort to direct policy search methods, which are model-free in that they can learn an optimal policy without knowledge of the environment. In this section, we lay the groundwork for these model-free methods by deriving an OIR policy gradient theorem that allow us to sample from \( \nabla \rho(\theta) \), the gradient of \( \rho \) with respect to the policy parameters.

Recall the definition of the OIR given in (6):

\[
\rho(\theta) = \frac{J(\theta)}{\kappa + H(d_\theta)},
\]

where \( \kappa \geq 0 \) is some user-specified constant. We are interested in minimizing this objective by performing stochastic gradient descent in policy parameters \( \theta \). Doing so is not straightforward using existing tools, however, as obtaining stochastic estimates of \( \nabla \rho(\theta) \) involves estimating

\[
\nabla \rho(\theta) = \frac{\nabla J(\theta)(\kappa + H(d_\theta)) - J(\theta)\nabla H(d_\theta)}{[\kappa + H(d_\theta)]^2}.
\]

Though we can use the classical policy gradient theorem (to be stated in [10] and discussed below) to estimate \( \nabla J(\theta) \) and we can empirically estimate \( J(\theta) \) and \( H(d_\theta) \), it is not immediately clear how to estimate \( \nabla H(d_\theta) \).

In what follows we show how \( \nabla H(d_\theta) \) and consequently \( \nabla \rho(\theta) \) can be estimated. First, we recall some preliminary material from the classic policy gradient literature. Second, en route to proving our OIR policy gradient theorem, we derive tractable expressions for the gradient of the entropy of the state occupancy measure, \( \nabla H(\lambda_\theta) \), as well as the gradient of the entropy of the state-action occupancy measure, \( \nabla H(\lambda_\theta) \). These entropy gradient theorems are likely of independent interest. Last, we provide our OIR policy gradient theorem, which we use in the subsequent section to develop policy gradient algorithms for minimizing the OIR.

#### 5.2.1 Policy Gradient Preliminaries

Consider a finite, average-cost MDP \( (S,A,p,c) \) and parametrized policy class \( \{\pi_\theta\}_{\theta \in \Theta} \). Given a policy \( \pi_\theta \), two important objects that we will need in what follows are the relative state value function

\[
V_\theta(s) = \sum_{i=0}^\infty \mathbb{E}_{\pi_\theta}[c(s,a) - J(\theta) \mid s_0 = s]
\]

Theorem 1: Any optimal solution to the concave program \( (Q_3) \) is optimal for the OIR problem \( (Q) \).
and the relative action value function

\[ Q_\theta(s, a) = \sum_{t=0}^{\infty} \mathbb{E}_{\pi_\theta} [c(s, a) - J(\theta) \mid s_0 = s, a_0 = a], \]

where \( J(\theta) \) is as defined in (13).

Under the assumption that \( \pi_\theta(a|s) \) is differentiable in \( \theta \), for all \( s \in S, a \in A \), classic policy gradient methods minimize \( J(\theta) \) by iteratively moving along a stochastic descent direction with respect to policy parameters \( \theta \). The most fundamental approach is to simply perform gradient updates \( \theta \leftarrow \theta - \eta \nabla J(\theta) \), where \( \eta > 0 \) is a scalar stepsize. In order to carry out this update, it is necessary to estimate \( \nabla J(\theta) \) at each step. Fortunately, we are guaranteed by the policy gradient theorem [41] that, under certain conditions,

\[ \nabla J(\theta) = \sum_s d_\theta(s) \sum_a Q_\theta(s, a) \nabla \pi_\theta(a|s) = \sum_s \pi_\theta(a|s) Q_\theta(s, a) \nabla \log \pi_\theta(a|s) \]

\[ = \mathbb{E}_{\pi_\theta} \left[ Q_\theta(s, a) \nabla \log \pi_\theta(a|s) \right] \]

\[ = \mathbb{E}_{\pi_\theta} \left[ (c(s, a) - J(\theta)) \nabla \log \pi_\theta(a|s) \right]. \tag{10} \]

By simply following policy \( \pi_\theta \), we can sample from the right-hand side of (10) to obtain stochastic estimates of \( \nabla J(\theta) \), then use them to minimize \( J(\theta) \) via stochastic gradient descent. This expression will prove useful for obtaining the gradient of the OIR in (9).

5.2.2 The Entropy Gradient

As explained above, to estimate \( \nabla \rho(\theta) \) we must know how to estimate \( \nabla H(d_\theta) \). Fortunately, by using the relationship between entropy and cross-entropy and leveraging the existing policy gradient theorem (10), \( \nabla H(d_\theta) \) can be estimated in a straightforward manner.

Given two policy parameters \( \theta \) and \( \theta' \) and the corresponding state occupancy measures \( d_\theta, d_{\theta'} \) [cf. (1)], recall the definition of state occupancy measure entropy \( H(d_\theta) \) given in (4). Furthermore, define

\[ CE(d_\theta, d_{\theta'}) = -\sum_s d_\theta(s) \log d_{\theta'}(s), \tag{11} \]

\[ D_{KL}(d_\theta \parallel d_{\theta'}) = \sum_s \log \left( \frac{d_\theta(s)}{d_{\theta'}(s)} \right) d_\theta(s), \tag{12} \]

the cross-entropy and Kullback-Leibler (KL) divergence between \( d_\theta \) and \( d_{\theta'} \), respectively. Also recall the useful fact that

\[ CE(d_\theta, d_{\theta'}) = H(d_\theta) + D_{KL}(d_\theta \parallel d_{\theta'}). \]

We now show that, when evaluated at \( \theta = \theta' \), the gradients of the entropy and cross entropy are equal.

Lemma 4. For any \( \theta' \in \Theta \),

\[ \nabla H(d_\theta)|_{\theta = \theta'} = \nabla CE(d_\theta, d_{\theta'})|_{\theta = \theta'}. \tag{13} \]

Proof. Notice that

\[ \nabla CE(d_\theta, d_{\theta'})|_{\theta = \theta'} = \nabla \left[ H(d_\theta) + D_{KL}(d_\theta \parallel d_{\theta'}) \right]|_{\theta = \theta'} \]

\[ = \nabla H(d_\theta)|_{\theta = \theta'} + \nabla D_{KL}(d_\theta \parallel d_{\theta'})|_{\theta = \theta'}. \tag{14} \]

\[ ^5 \]In what follows, for a given \( f : \Theta \rightarrow \mathbb{R} \), we will use the notation \( \nabla f(\theta)|_{\theta = \theta_1} \) to emphasize the fact that the gradient of \( f \) with respect to (w.r.t.) \( \theta \) is being taken first, then subsequently evaluated at \( \theta = \theta_1 \). When there is no ambiguity, however, we will revert to the standard practice of using \( \nabla f(\theta_1) \) to denote \( \nabla f(\theta)|_{\theta = \theta_1} \), where convenient.
Lemma 4 gives us the following important fact:

Expanding the term $\nabla D_{KL}(d_\theta \parallel d_{\theta'})$, where the gradient is being taken w.r.t. $\theta$ and $\theta'$ is fixed, we get

$$\nabla D_{KL}(d_\theta \parallel d_{\theta'}) = \sum_s \nabla \left[ \log \left( \frac{d_\theta(s)}{d_{\theta'}(s)} \right) \right] d_\theta(s) = \sum_s \nabla \left[ (\log d_\theta(s) - \log d_{\theta'}(s))d_\theta(s) \right]$$

(a) $= \sum_s \left[ (\nabla \log d_\theta(s) - \log d_{\theta'}(s)) \right] d_\theta(s) + \left[ \log d_\theta(s) - \log d_{\theta'}(s) \right] \nabla d_\theta(s)$

(b) $= \sum_s \left[ \frac{\nabla d_\theta(s)}{d_\theta(s)} \right] d_\theta(s) + \left[ \log d_\theta(s) - \log d_{\theta'}(s) \right] \nabla d_\theta(s)$

(c) $= \sum_s \left[ 1 + \log d_\theta(s) - \log d_{\theta'}(s) \right] \nabla d_\theta(s)$,

where (a) follows from the product rule, (b) holds because $\nabla \log d_\theta(s) = \left[ \nabla d_\theta(s) \right] / d_\theta(s)$, and the $d_\theta(s)$ terms cancel to yield (c). Evaluating the last expression at $\theta = \theta'$, we end up with

$$\nabla D_{KL}(d_\theta \parallel d_{\theta'})|_{\theta = \theta'} = \left( \sum_s \left[ 1 + \log d_\theta(s) - \log d_{\theta'}(s) \right] \nabla d_\theta(s) \right)|_{\theta = \theta'}$$

(a) $= \sum_s \left[ 1 + \log d_{\theta'}(s) - \log d_{\theta'}(s) \right] \left( \nabla d_\theta(s) \right)|_{\theta = \theta'}$

(b) $= \sum_s \left( \nabla d_\theta(s) \right)|_{\theta = \theta'} = \left( \sum_s \nabla d_\theta(s) \right)|_{\theta = \theta'} = \left( \nabla \sum_s d_\theta(s) \right)|_{\theta = \theta'}$

(c) $= \nabla 1|_{\theta = \theta'} = 0$,

where we evaluate the terms $\log d_\theta(s)$ and $\nabla d_\theta(s)$ at $\theta = \theta'$ to get (a), move the evaluation $\theta = \theta'$ outside the summation in (b), pull the gradient outside the summation to obtain (c), and finally recall that $\sum_s d_\theta(s) = 1$ to get (d). But, recalling equation (14), this means that

$$\nabla H(d_\theta)|_{\theta = \theta'} = \nabla CE(d_\theta, d_{\theta'})|_{\theta = \theta'},$$

completing the proof.

Lemma 4 gives us the following important fact:

We can estimate the entropy gradient $\nabla H(d_\theta)|_{\theta = \theta_i}$ by instead estimating the cross-entropy gradient $\nabla CE(d_\theta, d_{\theta_i})|_{\theta = \theta_i}$.

At first glance, this simply substitutes one problem for another. However, given a fixed $\theta_i$, for any $\theta$, we can use the classic policy gradient theorem (10) to obtain a tractable expression for $\nabla CE(d_\theta, d_{\theta_i})|_{\theta = \theta_i}$ that can be estimated via Monte Carlo, as described next.

### 5.2.3 Entropy and OIR Policy Gradient Theorems

We now provide three new policy gradient results. The first two provide tractable expressions for the gradient of the state occupancy measure entropy, $\nabla H(d_\theta)$, and the gradient of the state-action occupancy measure entropy, $\nabla H(\lambda_\theta)$, while the third provides a tractable expression for $\nabla \rho(\theta)$. Together, these results unlock policy gradient algorithms for maximizing the entropies $H(d_\theta)$ and $H(\lambda_\theta)$ and minimizing the OIR, $\rho(\theta)$.

**Theorem 2.** Let an MDP $(S, A, p, c)$ and a differentiable parametrized policy class $\{\pi_\theta\}_{\theta \in \Theta}$ be given, and recall the definition (1) of the state occupancy measure $d_\theta$ induced by each $\pi_\theta$ on $S$. Fix a policy parameter iterate $\theta_i$ at timestep $t$. The gradient $\nabla H(d_\theta)|_{\theta = \theta_i}$ [cf. (13)] with respect to the policy parameters $\theta$ of the state occupancy measure entropy $H(d_\theta)$ [cf. (4)] evaluated at $\theta = \theta_i$ satisfies

$$\nabla H(d_\theta)|_{\theta = \theta_i} = \mathbb{E}_{\pi_\theta_i} \left[ (-\log d_{\theta_i}(s) - H(d_{\theta_i})) \nabla \log \pi_{\theta_i}(a|s) \right].$$

(15)
Proof. At a given $t$, consider the average-reward MDP $({\mathcal S}, {\mathcal A}, p, r_t)$, where $r_t : {\mathcal S} \to \mathbb{R}^+$ is the purely state-dependent (i.e., action-independent) reward given by $r_t(s) = -\log d_{\theta_t}(s)$. We refer to the MDP $({\mathcal S}, {\mathcal A}, p, r_t)$ as the shadow MDP associated with parameter $\theta_t$. Define

$$
\mu^t(\theta) = \lim_{n \to \infty} \frac{1}{n} \mathbb{E} \left[ \sum_{i=1}^{n} r_i(s_i) \mid \pi_{\theta} \right],
$$

$$
Q^t_\theta(s,a) = \sum_{n=1}^{\infty} \mathbb{E} \left[ r_t(s_n) - \mu^t(\theta) \mid s_0 = s, a_0 = a, \pi_{\theta} \right].
$$

Here $\mu^t(\theta)$ is the expected long-run average reward of using $\pi_{\theta}$ on the shadow MDP $({\mathcal S}, {\mathcal A}, p, r_t)$, while $Q^t_\theta$ is the corresponding state-action value function. Given a policy $\pi_{\theta}$,

$$
CE(d_{\theta}, d_{\theta_t}) = -\sum_s d_{\theta}(s) \log d_{\theta_t}(s) = \sum_s d_{\theta}(s) r_t(s) = \sum_s d_{\theta}(s) \sum_a \pi_{\theta}(a|s) r_t(s),
$$

where the last equality holds by the fact that $\sum_a \pi_{\theta}(a|s) = 1$, for all $s \in {\mathcal S}$. The policy gradient theorem \cite{1} can thus be used to yield the expression

$$
\nabla CE(d_{\theta}, d_{\theta_t}) = \sum_s d_{\theta}(s) \sum_a Q^t_\theta(s,a) \nabla \log \pi_{\theta}(a|s).
$$

Combining expression \cite{17} with Lemma \cite{4} and noticing that $\mu^t(\theta_t) = \sum_s d_{\theta_t}(s)(-\log d_{\theta_t}(s)) = H(d_{\theta_t})$ yields equation \cite{15}.

We also have the following corollary.

**Corollary 1.** Let an MDP $({\mathcal S}, {\mathcal A}, p, c)$ and a differentiable parametrized policy class $\{\pi_{\theta}\}_{\theta \in \Theta}$ be given, and recall the definition \cite{2} of the state-action occupancy measure $\lambda_{\theta}$ induced by each $\pi_{\theta}$ on $\mathcal{S} \times \mathcal{A}$. Fix a policy parameter iterate $\theta_t$ at timestep $t$. The gradient $\nabla H(\lambda_{\theta})|_{\theta = \theta_t}$ with respect to the policy parameters $\theta$ of the state-action occupancy measure entropy $H(\lambda_{\theta})$ \cite{15} evaluated at $\theta = \theta_t$ satisfies

$$
\nabla H(\lambda_{\theta})|_{\theta = \theta_t} = \mathbb{E}_{\pi_{\theta_t}} \left[ (-\log d_{\theta}(s) - \log \pi_{\theta_t}(a|s) - H(\lambda_{\theta_t})) \nabla \log \pi_{\theta_t}(a|s) \right].
$$

**Proof.** The statement follows by replacing $r_t(s) = -\log d_{\theta_t}(s)$ with $r_t(s, a) = -\log \lambda_{\theta}(s, a) = -\log d_{\theta_t}(s) - \log \pi_{\theta_t}(a|s)$ in the proof of Theorem \cite{2} redefining $\mu^t(\theta)$ and $Q^t_\theta$ accordingly, and noting that

$$
CE(\lambda_{\theta}, \lambda_{\theta_t}) = -\sum_s \sum_a \lambda_{\theta}(s, a) \log \lambda_{\theta_t}(s, a) = \sum_s d_{\theta}(s) \sum_a \pi_{\theta}(a|s) [-\log d_{\theta_t}(s) - \log \pi_{\theta_t}(a|s)].
$$

An application of \cite{10} then yields the corresponding analogue of \cite{17}, which can be combined with a state-action analogue of Lemma \cite{4} to yield \cite{18}.

**Remark 2.** In this work we are interested in estimating $\nabla H(d_{\theta})$ because it is essential for estimating $\nabla \rho(\theta)$ \cite{9}. It is important to note, however, that Theorem \cite{2} and Corollary \cite{1} are of independent interest. Recent works \cite{15, 23, 20} consider the problem of maximizing entropy of occupancy measures, but none of them provide a simple gradient expression that can be used to directly perform gradient ascent in $\theta$ on the entropies \cite{11, 5}. Our entropy gradient theorem and corresponding corollary resolve this issue, and entropy maximization algorithms using it to maximize \cite{4, 5} are provided in the appendix. In addition to pure entropy maximization, Theorem \cite{2} and Corollary \cite{1} can be combined with standard policy gradient schemes to develop algorithms that incorporate a state or state-action exploration bonus, such as by minimizing $J(\theta) - H(d_{\theta})$ or $J(\theta) - H(\lambda_{\theta})$.

Now that we have a tractable expression for the entropy gradient $\nabla H(d_{\theta})$, we are ready to present our OIR policy gradient theorem:

---

\[ \text{Page 16} \]
Theorem 3. Let an MDP $(\mathcal{S}, \mathcal{A}, p, c)$, a differentiable parametrized policy class $\{\pi_\theta\}_{\theta \in \Theta}$, and a constant $\kappa \geq 0$ be given, and recall the definitions of the average cost $J(\theta)$ from equation (3), state occupancy measure $d_\theta$ from (1), and entropy $H(d_\theta)$ from (4). Fix a policy parameter iterate $\theta_t$ at timestep $t$. The gradient $\nabla \rho(\theta_t)$ [cf. (9)] with respect to the policy parameters $\theta$ of the OIR $\rho(\theta)$ [cf. (6)] evaluated at $\theta = \theta_t$ satisfies

$$\nabla \rho(\theta_t) = \mathbb{E}_{s,a} \left[ \frac{(c(s,a) - J(\theta_t)) (\kappa + H(d_\theta)) - J(\theta_t)(-\log d_\theta(s) - H(d_\theta))}{\kappa + H(d_\theta)} \nabla \log \pi_\theta(a|s) \right]$$

(19)

Proof. The claim follows by combining equations (9) and (10) with Theorem 2.

In light of Theorem 3 it is now possible to develop policy gradient algorithms for minimizing the OIR, to which we turn our attention next.

6 Algorithms

In this section we derive two policy gradient algorithms for minimizing the occupancy information ratio (9). The first, based on the classic REINFORCE algorithm, performs direct policy search, using Monte Carlo rollouts to estimate the gradient (19) of Theorem 3. The second is an actor-critic scheme with two critics: the standard critic corresponding to average cost, and an entropy critic corresponding to the shadow MDPs discussed in Theorem 2. Throughout this section, we will assume that an average-cost MDP $(\mathcal{S}, \mathcal{A}, p, c)$ has been given. The algorithms that we present will minimize (6) over this underlying MDP.

6.1 Information-Directed REINFORCE

The classic REINFORCE algorithm [46] generates a single, finite trajectory using a fixed policy, estimates the corresponding policy gradient based on the trajectory, and updates the policy parameters based on the policy gradient estimate. However, it cannot apply to minimizing (6) due to the presence of the entropy of the occupancy measure induced by policy $\pi_\theta$. Thus, we develop a REINFORCE-type algorithm, which we call Information-Directed REINFORCE (ID-REINFORCE), which uses a similar procedure to estimate (19) and perform policy updates.

At a high level, the operation of the ID-REINFORCE algorithm for minimizing (6) is straightforward. At each timestep $t$, the algorithm generates a trajectory using the current policy $\pi_\theta$. It then forms estimates of $J(\theta_t)$ and $H(d_\theta)$ defined in (3) and (4), respectively, and in turn uses these to estimate $\nabla \rho(\theta_t)$ by leveraging (19). This gradient estimate is then used to update the policy parameters. Note that, in order to estimate $H(d_\theta)$, it is necessary to first estimate $d_\theta$. For ease of exposition, we assume access to an oracle DensityEstimator that returns the occupancy measure $d_\theta = \text{DensityEstimator}(\theta)$ when provided with input policy parameter $\theta \in \Theta$. When $\mathcal{S}$ is finite and not too large, DensityEstimator can be implemented by computing the empirical visitation probabilities for each of the states $s \in \mathcal{S}$ based on sample trajectories. We restrict our attention to this setting in this paper. When $\mathcal{S}$ is large or continuous, on the other hand, various parametric and nonparametric density estimation techniques can be used to implement DensityEstimator. Such extensions to continuous spaces are an important direction for future work.

A more detailed description of the operation of ID-REINFORCE is as follows. Fix a parametric policy class $\{\pi_\theta\}_{\theta \in \Theta}$, a step-size sequence $\{\eta_t\}_{t \in \mathbb{N}}$, and initial policy parameters $\theta_0$. At each time-step $t \geq 0$, generate a trajectory $\{(s_i, a_i)\}_{i=1}^K$ of length $K$ using policy $\pi_{\theta_t}$, and call $d_{\theta_t} = \text{DensityEstimator}(\theta_t)$. Construct the estimates

$$\hat{J}(\theta_t) = \frac{1}{K} \sum_{i=1}^K c(s_i, a_i) \approx J(\theta_t), \quad \hat{H}(d_{\theta_t}) = \frac{1}{K} \sum_{i=1}^K (-\log d_{\theta_t}(s_i)) \approx H(d_{\theta_t}).$$

Using these, form the following estimate of $\nabla \rho(\theta_t)$:

$$\nabla \rho(\theta_t) = \frac{1}{K} \sum_{i=1}^K \left[ \frac{(c(s_i, a_i) - \hat{J}(\theta_t)) (\kappa + \hat{H}(d_{\theta_t})) - \hat{J}(\theta_t)(-\log d_{\theta_t}(s_i) - \hat{H}(d_{\theta_t}))}{\kappa + \hat{H}(d_{\theta_t})} \nabla \log \pi_{\theta_t}(a_i|s_i) \right] \nabla \log \pi_{\theta_t}(a_i|s_i).$$
Next, perform the policy gradient update $\theta_{t+1} \leftarrow \theta_t + \eta_t \nabla \rho(\theta_t)$. Increment $t \leftarrow t + 1$ and repeat until convergence.

Instability can arise from estimating $\hat{J}(\theta_t)$ and $\hat{H}(d_{\theta_t})$ based on a single trajectory, however. To mitigate this, we can maintain the following exponential moving averages across trajectories:

$$\tilde{\mu}_t^J = (1 - \tau_t)\tilde{\mu}_{t-1}^J + \tau_t \hat{J}(\theta_t), \quad \tilde{\mu}_t^H = (1 - \tau_t)\tilde{\mu}_{t-1}^H + \tau_t \hat{H}(d_{\theta_t}),$$

where $\{\tau_t\}_{t \in \mathbb{N}} \subset (0, 1]$ is a given (typically constant) step-size sequence. Notice that when $\tau_t = 1$, for all $t$, we recover the gradient expression above. Pseudocode for ID-REINFORCE with exponential moving averages and constant step-sizes is given in Algorithm 1.

Algorithm 1: ID-REINFORCE

Input: Rollout length $K$, step-sizes $\eta > 0$, $\tau \in (0, 1]$, parametrized policy class $\{\pi_\theta\}_{\theta \in \Theta}$, entropy additive constant $\kappa \geq 0$

1. Initialization: randomly sample $\theta_0$; select $\tilde{\mu}_0^H, \tilde{\mu}_0^J > 0$; $t \leftarrow 0$

2. while not converged do

3. Generate trajectory $\{(s_i, a_i)\}_{i=1,...,K}$ using policy $\pi_{\theta_t}$

4. $\hat{J}(\theta_t) = \frac{1}{K} \sum_{i=1}^{K} c(s_i, a_i)$

5. $\tilde{\mu}_t^J = (1 - \tau_t)\tilde{\mu}_{t-1}^J + \tau_t \hat{J}(\theta_t)$ \hspace{1cm} \textcircled{Update $J(\theta)$ estimate}

6. $d_{\theta_t} = \text{DENSITYESTIMATOR}(\theta_t)$ \hspace{1cm} \textcircled{Call to oracle}

7. $\hat{H}(d_{\theta_t}) = \frac{1}{K} \sum_{i=1}^{K} (-\log d_{\theta_t}(s_i))$

8. $\tilde{\mu}_t^H = (1 - \tau_t)\tilde{\mu}_{t-1}^H + \tau_t \hat{H}(d_{\theta_t})$ \hspace{1cm} \textcircled{Update $H(d_{\theta})$ estimate}

9. $\nabla \rho(\theta_t) = \frac{1}{\kappa + \tilde{\mu}_t^H} \frac{1}{K} \sum_{i=1}^{K} \left[ \left( c(s_i, a_i) - \tilde{\mu}_t^J \right) \left( \kappa + \tilde{\mu}_t^H \right) - \tilde{\mu}_t^J \left( -\log d_{\theta_t}(s_i) - \tilde{\mu}_t^H \right) \right] \nabla \log \pi_{\theta_t}(a_i | s_i)$

10. $\theta_{t+1} = \theta_t - \eta_t \nabla \rho(\theta_t)$ \hspace{1cm} \textcircled{Policy gradient update}

11. $t \leftarrow t + 1$

12. end

6.2 Information-Directed Actor-Critic

In practice, REINFORCE-type algorithms can suffer from numerical instability, since the Monte Carlo estimates of the state values can have high variance. Actor-critic algorithms, on the other hand, use a value function estimator to provide lower-variance estimates of these quantities in exchange for increased bias, often resulting in improved empirical performance. In this section we develop the Information-Directed Actor-Critic (IDAC) algorithm, a variant of the classic actor-critic algorithm [21, 10] with two critics: the standard critic corresponding to average cost [3], and an entropy critic corresponding to the shadow MDPs $\langle S, A, p, r_t \rangle$, $t \geq 0$, where $r_t(s, a) = -\log d_{\theta_t}(s)$ is the shadow reward defined in Theorem 2. As in the discussion of the ID-REINFORCE algorithm above, we assume access to the DENSITYESTIMATOR oracle throughout.

In broad strokes, the classic actor-critic algorithm for minimizing [3] alternates between critic and actor updates. At each timestep, it first computes the temporal difference (TD) error, which is a bootstrapped estimate of the amount by which the current state value function approximator, known as the critic, over- or underestimates the true value of the current state (see [10] for details). This TD error is then used to update the critic, which is in turn used to update the policy, or actor.

More specifically, given a parametric policy class $\{\pi_\theta\}_{\theta \in \Theta}$ and a class $\{v_\omega\}_{\omega \in \Omega}$ of state value function approximators, the classic actor-critic scheme for minimizing $J(\theta)$ is as follows. First, positive step-size sequences $\{\alpha_t\}_{t \in \mathbb{N}}, \{\beta_t\}_{t \in \mathbb{N}}$, and $\{\tau_t\}_{t \in \mathbb{N}}$ satisfying certain conditions are specified, initial parameters $\theta_0$ and $\omega_0$ are chosen, and a scalar $\mu_{-1} > 0$ is initialized. At time-step $t \geq 0$, the exponential moving average $\mu_t$ maintaining a running estimate of $J(\theta_t)$ is updated via

$$\mu_t = (1 - \tau_t)\mu_{t-1} + \tau_t c(s_t, a_t),$$

(20)
the TD error
\[ \delta_t = c(s_t, a_t) - \mu_t + v_{\omega_t}(s_{t+1}) - v_{\omega_t}(s_t), \] (21)
is computed, and the value function update
\[ \omega_{t+1} = \omega_t + \alpha_t \delta_t \nabla v_{\omega_t}(s_t) \] (22)
and policy update (with baseline; see, e.g., [10, 40])
\[ \theta_{t+1} = \theta_t - \beta_t \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \] (23)
are performed. Note that the update (23) corresponds to taking a gradient descent step in the direction \( \nabla J(\theta_t) \).

For our IDAC algorithm, we modify the classic scheme just discussed by: (i) introducing an additional \textit{entropy critic} allowing us to estimate the entropy gradient given in Theorem 2, and (ii) altering the policy update to take a gradient descent step in the direction \( \nabla \rho(\theta_t) \) instead of \( \nabla J(\theta_t) \). We first summarize the key steps of IDAC. It begins each timestep by computing two different TD errors: one corresponds to the critic for the MDP \((S,A,p,c)\), which we call the \textit{cost critic}, while the other corresponds to the critic for the shadow MDP \((S',A,p',r_1)\), which we call the \textit{entropy critic}. Note that the entropy critic TD error computation requires a call to \textsc{DensityEstimator}. Next, the cost and entropy critic TD errors are used to update their respective critics, which are in turn combined to perform the actor update.

A detailed description of IDAC is as follows. Let a parameterized policy class \( \{\pi_{\theta}\}_{\theta \in \Theta} \) and a class \( \{v_{\omega}\}_{\omega \in \Omega} \) of state value function approximators be given. We assume for ease of exposition that the same class \( \{v_{\omega}\}_{\omega \in \Omega} \) of function approximators is used for both the cost and entropy critics, but this assumption can be removed both in practice and in the convergence analysis provided in the subsequent section. Initialize step-size sequences \( \{\alpha_t\}, \{\beta_t\}, \{\tau_t\} \) and policy parameter \( \theta_0 \) as above. Initialize the cost and entropy critic parameters \( \omega_0^j \) and \( \omega_0^H \), respectively, as well as constants \( \mu_{-1}, \mu_{-1} > 0 \). At time-step \( t \), call \( d_{\theta_t} = \text{DensityEstimator}(\theta_t) \), then update the moving averages via

\[ \mu_t^j = (1 - \tau_t)\mu_{t-1}^j + \tau_t c(s_t, a_t), \] (24)
\[ \mu_t^H = (1 - \tau_t)\mu_{t-1}^H - \tau_t \log d_{\theta_t}(s_t). \] (25)

Next, update the cost and entropy TD errors
\[ \delta_t^j = c(s_t, a_t) - \mu_t^j + v_{\omega_t^j}(s_{t+1}) - v_{\omega_t^j}(s_t), \] (26)
\[ \delta_t^H = -\log d_{\theta_t}(s_t) - \mu_t^H + v_{\omega_t^H}(s_{t+1}) - v_{\omega_t^H}(s_t), \] (27)
and perform the corresponding critic updates
\[ \omega_{t+1}^j = \omega_t^j + \alpha_t \delta_t^j \nabla v_{\omega_t^j}(s_t), \] (28)
\[ \omega_{t+1}^H = \omega_t^H + \alpha_t \delta_t^H \nabla v_{\omega_t^H}(s_t). \] (29)

Finally, carry out the actor update
\[ \theta_{t+1} = \theta_t - \beta_t \frac{\delta_t^j (\mu_t^H - \mu_t^j) - \mu_t^j \delta_t^H}{(\mu_t^H)^2} \nabla \log \pi_{\theta_t}(a_t|s_t). \] (30)

The iterative algorithm just given is what we will study in our convergence analysis. In practice, however, it is often advantageous to perform rollouts of some length \( K \) at each time-step, similar to ID-REINFORCE above, then use data from the entire rollout to obtain lower-variance gradient estimates for the critic and actor updates. Pseudocode for IDAC with length-\( K \) rollouts and constant step-sizes is provided in Algorithm 2.
In this subsection we prove that the OIR optimization problem enjoys a powerful suitable conditions. Taken together, these results prove that both algorithms converge to globally optimal solutions under suitable conditions. This result is surprising, as the objective function of the optimization problem

\[ \min_{\theta \in \Theta} \rho(\theta) = \frac{J(\theta)}{\kappa + H(d_\theta)}. \]  

This result is surprising, as the objective function \( \rho(\theta) \) is highly non-convex.

Let \( \Theta \subseteq \mathbb{R}^k \) be a convex set of permissible policy parameters and let a parametrized policy class \( \{\pi_\theta\}_{\theta \in \Theta} \) be given. Let \( \lambda : \Theta \rightarrow \mathcal{D}(S \times A) \) be a function mapping each parameter vector \( \theta \in \Theta \) to the state-action occupancy measure \( \lambda(\theta) := \lambda_\theta := \lambda_{\pi_\theta} \) induced by the policy \( \pi_\theta \) over \( S \times A \). We make the following assumptions.

6 Though we use all three definitions of the state-action occupancy measure interchangeably, we will typically use \( \lambda(\cdot) \) as a function (e.g., when taking gradients of it), \( \lambda_\theta \) to emphasize the dependence of the occupancy measure on the policy parameter \( \theta \), and \( \lambda_{\pi_\theta} \) to stress its dependence on the policy \( \pi_\theta \).
Assumption 2. The set of permissible policy parameters $\Theta$ is a compact set. Furthermore, for any $s \in S, a \in A$, the function $\pi_\theta(a|s)$ is continuously differentiable with respect to $\theta$ on $\Theta$, and the Markov chain induced by $\pi_\theta$ on $S$ is ergodic.

Assumption 3. Suppose the following statements hold:

1. $\lambda(\cdot)$ gives a bijection between $\Theta$ and its image $\lambda(\Theta)$, and $\lambda(\Theta)$ is compact and convex.
2. Let $h(\cdot) := \lambda^{-1}(\cdot)$ denote the inverse mapping of $\lambda(\cdot)$. $h(\cdot)$ is Lipschitz continuous.
3. The Jacobian matrix $\nabla \lambda(\theta)$ is Lipschitz on $\Theta$.

Assumption 2 is standard in the policy gradient literature, and it implies that $\nabla \rho(\theta)$ exists, for all $\theta \in \Theta$. Also, Assumption 3 holds in certain settings, as illustrated in the following example.

Example 1. Consider an MDP with state space $S = \{s_1, s_2\}$, action space $A = \{L, R\}$, a transition probability function $p$ satisfying $p(s'|s, a) > 0$, for all $s, s' \in S, a \in A$, and an arbitrary cost function $c : S \times A \rightarrow \mathbb{R}$. Given a policy $\pi$, we can represent it in tabular form as a vector $\pi \in \mathbb{R}^4$ with non-negative entries such that $\pi(L|s_i) + \pi(R|s_i) = 1$, for $i \in \{1, 2\}$. The transition probability matrix of the Markov chain induced by $\pi$ on $S$ is as follows:

$$P_\pi = \begin{bmatrix}
    p(s_1|s_1, L)\pi(L|s_1) + p(s_1|s_1, R)\pi(R|s_1) & p(s_2|s_1, L)\pi(L|s_1) + p(s_1|s_1, R)\pi(R|s_1) \\
    p(s_1|s_2, L)\pi(L|s_2) + p(s_1|s_2, R)\pi(R|s_2) & p(s_2|s_2, L)\pi(L|s_2) + p(s_2|s_2, R)\pi(R|s_2)
\end{bmatrix}.$$  

The state occupancy measure $d_\pi$ induced by $\pi$ can be obtained by solving the system of equations $P_\pi^T \cdot d = d, 1^T \cdot d = 1$ in the unknown $d = [d_1, d_2]^T$. Some simple algebra yields

$$d_1 = \frac{p(s_1|s_2, L)\pi(L|s_2) + p(s_1|s_2, R)\pi(R|s_2)}{1 + p(s_1|s_2, L)\pi(L|s_2) + p(s_1|s_2, R)\pi(R|s_2) - p(s_1|s_1, L)\pi(L|s_1) - p(s_1|s_1, R)\pi(R|s_1)}.$$  

Notice that, since $p(s_i|s_j, a) > 0$, $\pi(a|s_1) \geq 0$ and $\pi(L|s_i) + \pi(R|s_i) = 1$, for all $i, j \in \{1, 2\}$ and $a \in \{L, R\}$, $d_1$ is always defined and strictly positive. Since $d_2 = 1 - d_1$, the same statement holds for $d_2$. The state occupancy measure induced by $\pi$ is thus given by the vector $d_\pi = d = [d_1, d_2]^T$.

We now show that, for the MDP $(S, A, p, c)$ just specified, Assumption 3 holds. Let $\Theta \subset \mathbb{R}^4$ denote the set of vector representations of all valid policies $\pi$, and let $\lambda : \Theta \rightarrow \mathcal{D}(S \times A)$ denote the function mapping policies $\pi$ to state-action occupancy measures $\lambda(\pi)$. Clearly $\lambda(\pi)(s, a) = d_\pi(s)\pi(a|s)$, for all $s \in S, a \in A$. It is known that part one of Assumption 3 holds (see, e.g., [8]). It can furthermore be shown (see the proof of [43 Prop H.2]) that part two holds. All that remains is to show that part three holds by proving that $\nabla \lambda(\pi)$ is Lipschitz in $\pi$. Notice that the identity function is Lipschitz and trivially bounded over any bounded domain, and recall that the product of two Lipschitz and bounded functions is a Lipschitz function. Since the domain $\Theta$ is bounded, this means that, if $d_\pi$ is Lipschitz and bounded in $\pi$, it will follow that $\lambda(\pi)$ is Lipschitz in $\pi$ over $\Theta$, since it is a product of Lipschitz, bounded functions. But the partial derivatives of $d_1$ with respect to $\pi(L|s_1), \pi(R|s_1), \pi(L|s_2), \pi(R|s_2)$ are all continuous and bounded for all valid policy vectors $\pi$, so $d_1$ is Lipschitz in $\pi$. Since $d_2 = 1 - d_1$, this means that $d_\pi = [d_1, d_2]^T$ is Lipschitz in $\pi$. Finally, we already know that $d_\pi$ is bounded, so $\lambda(\pi)$ is therefore Lipschitz in $\pi$, and thus part three of Assumption 3 holds.

The above example can likely be extended to prove that Assumption 3 holds in the tabular setting under suitable ergodic conditions on the underlying MDP.

We now have the following theorem, which is the main result of this subsection.

Theorem 4. Let Assumptions 2 and 3 hold. Let $\theta^*$ be a stationary point of (31), i.e.,

$$\nabla \rho(\theta^*) = 0.$$  

(32)

Then $\theta^*$ is globally optimal for (31).
Proof. The key idea behind the proof is to show that the stationary point \( \theta^* \) corresponds to an optimal solution to the concave program (34), and thus also provides an optimal solution to the quasiconvex OIR minimization problem (31). To do this we first reinterpret (31) as a concave program via the perspective transformation considered in section 5.1. We then demonstrate that the feasible solution corresponding to \( \theta^* \) is a stationary point of this concave program and thus a global optimal solution of the problem. This will imply that \( \theta^* \) is optimal for (31). The proof builds on that of [48, Thm. 4.2], with key modifications to accommodate the fact that the underlying OIR optimization problem is not convex, but quasiconvex in the state-action occupancy measure.

We first reformulate (31) as a maximization problem. Let \( q(\theta) = 1/\rho(\theta) = (\kappa + H(d_0))/J(\theta) \). Notice that \( \hat{H}(\lambda_0) = H(d_0), \) where \( H(\lambda) = H(d^\lambda) \) is the entropy of the state occupancy measure \( d^\lambda \in D(S) \) given by \( d^\lambda = \sum_{s} \lambda_s \). Also recall that \( J(\theta) = c^T \lambda_\theta \), for some vector \( c > 0 \) of costs. Together, this means that \( q(\theta) = (\kappa + \hat{H}(\lambda_0))/J(\lambda_0) \).

In what follows we prove that \( \theta^* \) is globally optimal for \( \max_{\theta \in \Theta} q(\theta) \). By Lemma 3, this will imply that \( \theta^* \) is globally optimal for \( \min_{\theta \in \Theta} \rho(\theta) \). A first important fact to notice is that, since \( \rho(\theta) \) is strictly positive on \( \Theta \), we know \( q(\theta) \) is differentiable in \( \theta \) and \( \nabla q(\theta) = -\nabla \rho(\theta)/[\rho(\theta)]^2 \), for all \( \theta \in \Theta \). By (32), this means \( \nabla q(\theta^*) = 0 \), so \( \theta^* \) is a stationary point of the optimization problem \( \max_{\theta \in \Theta} q(\theta) \).

Keeping this in mind, we now reinterpret the problem \( \max_{\theta \in \Theta} q(\theta) \) as a related concave program. For \( z \in \mathbb{R}^{\mid S \mid \mid A \mid + 1} \), let the vector \( y \in \mathbb{R}^{\mid S \mid \mid A \mid} \) denote all but the last entry in \( z \), and let the scalar \( t \) denote the last entry of \( z \). We will write \( z = (y,t) \) for brevity. Let \( \zeta: \mathcal{D}(S \times A) \to \mathbb{R}^{\mid S \mid \mid A \mid + 1} \) be the mapping given by \( \zeta(\lambda) = (\lambda/J(\lambda), 1/J(\lambda)) \). Consider the optimization problems

\[
\max_{\lambda \in \lambda(\Theta)} \frac{\kappa + \hat{H}(\lambda)}{J(\lambda)} \tag{33}
\]

and

\[
\max_{z \in (\zeta \circ \lambda)(\Theta)} P_{\kappa, \hat{H}}(z), \tag{34}
\]

where \( P_{\kappa, \hat{H}}: \mathbb{R}^{\mid S \mid \mid A \mid + 1} \to \mathbb{R} \) denotes the perspective transformation of \( \kappa + \hat{H}(\lambda) \), given by \( P_{\kappa, \hat{H}}(z) = P_{\kappa, \hat{H}}((y,t)) = t(\kappa + \hat{H}(y,t)) \). For notational convenience we henceforth drop the dependency on \( \kappa \) and simply write \( P_{\hat{H}} \) and \( \hat{H} \) instead of \( P_{\kappa, \hat{H}} \) and \( \kappa + \hat{H} \). Recall that, since \( \hat{H} \) is concave over the region \( \mathcal{D}(S \times A) \), its perspective transform \( P_{\hat{H}} \) is concave over the region \( \zeta(\mathcal{D}(S \times A)) \). \( P_{\hat{H}} \) is thus concave over the convex, compact region \( (\zeta \circ \lambda)(\Theta) \subseteq (\zeta(\mathcal{D}(S \times A)) \).

Our goal is to show that \( \theta^* \) corresponds to a stationary point of the concave program (34). To see why this suffices, consider the following. Clearly if \( \lambda^* = \lambda(\theta^*) \) is globally optimal for (33), then \( \theta^* \) is globally optimal for (31). Notice that the problem (33) is quasiconcave, as the objective is quasiconcave and the feasible region is convex and compact. Furthermore, since \( \hat{H} \) is concave and \( J \) is linear and strictly positive in \( \lambda \), the optimization problem (34) is the concave transformation of problem (33) by the discussion in Section 5.1. This means that any optimal solution \( z^* \) of (34) is both globally optimal for (34) and allows us to recover a solution \( \lambda^* = \zeta^{-1}(z^*) \) that is globally optimal for (33) and thus (31). To prove the theorem, it thus suffices to prove that \( z^* = (\zeta \circ \lambda)(\theta^*) \) is globally optimal for (34).

The remainder of the proof provides the technical details demonstrating that \( z^* \) is a stationary point of (34). We first show that the conditions of Assumption 3 can be extended to the mapping \( \zeta \circ \lambda \), then use these properties to make a series of bounding arguments that ultimately prove the stationarity of \( z^* \). In order to complete the first task of extending the conditions of Assumption 3 to \( \zeta \circ \lambda \), we need to prove the following:

(i) \( \zeta \circ \lambda \) gives a bijection between \( \Theta \) and \( (\zeta \circ \lambda)(\Theta) \);
(ii) \( \zeta \circ \lambda \) has a Lipschitz continuous inverse; and
(iii) the Jacobian \( \nabla_{\theta}(\zeta \circ \lambda)(\theta) \) is Lipschitz.

We first prove (i) and (ii). Given Assumption 3 this boils down to showing that \( \zeta(\lambda) = (\lambda/J(\lambda), 1/J(\lambda)) \) gives a bijection between \( \Theta \) and \( (\zeta \circ \lambda)(\Theta) \) and that \( \zeta \) has a Lipschitz continuous inverse. \( \zeta \) is a surjection onto \( (\zeta \circ \lambda)(\Theta) \) by definition, so we just need to show it is injective. Fix \( \lambda \neq \lambda' \). If \( J(\lambda) = J(\lambda') \), then \( \lambda/J(\lambda) = \lambda'/J(\lambda') \), so \( \zeta(\lambda) = \zeta(\lambda') \). If \( J(\lambda) \neq J(\lambda') \), on the other hand, then \( 1/J(\lambda) \neq 1/J(\lambda') \), so again \( \zeta(\lambda) \neq \zeta(\lambda') \). Therefore \( \zeta \) is injective and thus gives a bijection. The inverse of \( \zeta \) is clearly \( \zeta^{-1}(z) = \zeta^{-1}(y,t) = y/t \). Note that \( 0 < \min, c_i \leq t \leq \max, c_i < \infty \), so \( \zeta^{-1} \) has continuous, bounded partial derivatives and is thus Lipschitz continuous on \( (\zeta \circ \lambda)(\Theta) \). Combined with Assumption 3 this implies that \( \zeta \circ \lambda \) gives a bijection between \( \Theta \)
and \((\zeta \circ \lambda)(\Theta)\), proving (i). Since the composition of Lipschitz functions is Lipschitz, \(k = (\zeta \circ \lambda)^{-1} = \lambda^{-1} \circ \zeta^{-1}\) is Lipschitz continuous, proving (ii).

For (iii), an application of the chain rule gives \(\nabla_\theta (\zeta \circ \lambda)(\theta) = [\nabla_\lambda \zeta (\lambda(\theta))]^T \nabla_\theta \lambda(\theta)\). Clearly \(\nabla_\lambda \zeta(\lambda)\) is Lipschitz continuous and bounded over \(\Theta\), this implies that \(\nabla_\lambda \zeta(\lambda(\theta))\) is Lipschitz and bounded on \(\theta \in \Theta\). Furthermore, \(\nabla_\theta \lambda(\theta)\) is Lipschitz by assumption and bounded over \(\Theta\), so all entries in the matrix product \([\nabla_\lambda \zeta(\lambda(\theta))]^T \nabla_\theta \lambda(\theta)\) are sums and products of Lipschitz, bounded functions over \(\Theta\), implying that \(\nabla_\theta (\zeta \circ \lambda)(\theta)\) is Lipschitz on \(\Theta\), which proves (iii).

We now move on to the bounding arguments that will ultimately prove that \(z^*\) is a stationary point of \(\mathcal{P}\). First, notice that

\[
(P_H \circ \zeta \circ \lambda)(\theta) = P_H(\zeta(\lambda(\theta))) = \frac{\kappa + H(\lambda(\theta))}{J(\lambda(\theta))} = \frac{\kappa + \hat{H}(\lambda_0)}{J(\lambda_0)} = q(\theta),
\]

so \(\nabla_\theta (P_H \circ \zeta \circ \lambda)(\theta^*) = \nabla_\theta \frac{\kappa + \hat{H}(\lambda_0)}{J(\lambda_0)} = \nabla q(\theta^*) = 0\). Since \(P_H\) is concave and locally Lipschitz on \((\zeta \circ \lambda)(\Theta)\), by the chain rule we have

\[
\nabla_\theta (P_H \circ \zeta \circ \lambda)(\theta^*) = [\nabla_\theta (\zeta(\lambda(\theta^*)))^T \nabla_\theta P_H(z^*)] = 0,
\]

where \(z^* = (\zeta \circ \lambda)(\theta^*)\). This trivially implies that, for all \(\theta \in \Theta\),

\[
\langle \nabla_\theta P_H(z^*), \nabla_\theta (\zeta \circ \lambda)(\theta^* - \theta) \rangle = \langle [\nabla_\theta (\zeta(\lambda(\theta^*)))^T \nabla_\theta P_H(z^*)], \theta - \theta^* \rangle = 0. \tag{35}
\]

Equation (35) is important to the bounding arguments presented next.

In the following equations, let \(\theta = k(z)\) and \(\theta^* = k(z^*)\). Adding and subtracting \(\langle \nabla_\theta P_H(z^*), \nabla_\theta (\zeta \circ \lambda)(\theta^* - \theta) \rangle\), using equation (35), and applying the Cauchy-Schwarz inequality, we get

\[
\langle \nabla_\theta (\zeta \circ \lambda)(\theta), z - z^* \rangle = \langle \nabla_\theta P_H(z^*), (\zeta \circ \lambda)(\theta) - (\zeta \circ \lambda)(\theta^*) \rangle \tag{36}
\]

\[
= \langle \nabla_\theta P_H(z^*), \nabla_\theta (\zeta \circ \lambda)(\theta^*) \rangle \langle \theta - \theta^* \rangle + \langle \nabla_\theta P_H(z^*), (\zeta \circ \lambda)(\theta) - (\zeta \circ \lambda)(\theta^*) - \nabla_\theta (\zeta \circ \lambda)(\theta^*) \rangle \langle \theta - \theta^* \rangle \tag{37}
\]

\[
= \langle \nabla_\theta P_H(z^*), (\zeta \circ \lambda)(\theta) - (\zeta \circ \lambda)(\theta^*) - \nabla_\theta (\zeta \circ \lambda)(\theta^*) \rangle \langle \theta - \theta^* \rangle \tag{38}
\]

\[
\leq \left\| \nabla_\theta P_H(z^*) \right\| \left\| \nabla_\theta (\zeta \circ \lambda)(\theta) - (\zeta \circ \lambda)(\theta^*) - \nabla_\theta (\zeta \circ \lambda)(\theta^*) \right\|. \tag{39}
\]

Since \(\nabla_\theta (\zeta \circ \lambda)(\theta)\) is Lipschitz, there exists \(K_0 > 0\) such that

\[
\| \nabla_\theta (\zeta \circ \lambda)(\theta) - (\zeta \circ \lambda)(\theta^*) - \nabla_\theta (\zeta \circ \lambda)(\theta^*) \| \leq \frac{K_0}{2} \| \theta - \theta^* \|^2.
\]

In addition, \(k = (\zeta \circ \lambda)^{-1}\) is Lipschitz, so there exists \(K_1 > 0\) such that

\[
\| \theta - \theta^* \|^2 = \| k(z) - k(z^*) \|^2 \leq K_1^2 \| z - z^* \|^2.
\]

Combining these inequalities yields that

\[
\langle \nabla_\theta P_H(z^*), z - z^* \rangle \leq \frac{K_0 K_1^2}{2} \left\| \nabla_\theta P_H(z^*) \right\| \| z - z^* \|^2, \text{ for all } z \in (\zeta \circ \lambda)(\Theta). \tag{40}
\]

Since \((\zeta \circ \lambda)(\Theta)\) is convex, we can replace \(z\) above with \((1 - \alpha)z^* + \alpha z\) for any \(\alpha \in [0, 1]\), which gives

\[
\alpha \langle \nabla_\theta P_H(z^*), z - z^* \rangle \leq \frac{K_0 K_1^2 \alpha^2}{2} \left\| \nabla_\theta P_H(z^*) \right\| \| z - z^* \|^2, \text{ for all } z \in (\zeta \circ \lambda)(\Theta) \text{ and all } \alpha \in [0, 1].
\]

Dividing both sides by \(\alpha\) and taking the limit as \(\alpha\) approaches 0 from above, we obtain

\[
\langle \nabla_\theta P_H(z^*), z - z^* \rangle \leq 0, \text{ for all } z \in (\zeta \circ \lambda)(\Theta).
\]

Since problem \(\mathcal{P}\) is concave in \(z\), this implies that \(z^* = (\zeta \circ \lambda)(\theta^*)\) is a stationary point of that problem. The solution \(z^*\) is therefore a global optimal solution to \(\mathcal{P}\), which completes the proof. \(\Box\)
This powerful result guarantees that any stationary point of the policy optimization problem (31) is in fact a global optimum of that problem. In particular, this hidden quasiconcavity property implies that any policy gradient algorithm that can be shown to converge to a stationary point of the OIR optimization problem $\min_{\theta \in \Theta} \rho(\theta)$ in fact converges to a global optimum. Theorem 1 greatly strengthens the convergence results provided next by guaranteeing that they apply to global optima.

### 7.2 Non-Asymptotic Convergence Rate

Next, we establish a non-asymptotic convergence rate for the following projected gradient descent scheme for solving the OIR minimization problem (31):

$$\theta_{t+1} = \text{Proj}_\Theta (\theta_t - \eta \nabla \rho(\theta_t)) = \arg\min_{\theta} \left( \rho(\theta_t) + \langle \nabla \rho(\theta_t), \theta - \theta_t \rangle + \frac{1}{2\eta} \|\theta - \theta_t\|^2 \right),$$

(41)

for a fixed stepsize $\eta > 0$, where $\text{Proj}_\Theta$ denotes euclidean projection onto $\Theta$ and the rightmost equality holds by the convexity of $\Theta$. Note that the previous expression is a reformulation of Algorithm 1 with null gradient estimation error and projection onto the set $\Theta$ of permissible policy parameters; we assume the projection operation and that accurate gradient estimates are available for the purposes of analysis. We next formalize technical aspects of the setting.

Let $\Theta \subset \mathbb{R}^k$, $\{\pi_\theta\}_{\theta \in \Theta}$, and $\lambda : \Theta \rightarrow D(S \times A)$ be as in the previous section. Recall the mapping $\zeta : D(S \times A) \rightarrow [0,1]$ from the proof of Theorem 4, which was defined to be $\zeta(\lambda) = (\lambda/c^T \lambda, 1/c^T \lambda)$, where $c \in \mathbb{R}^{|S||A|}, c > 0$ is a vector of positive costs. Notice that, under the ergodicity conditions in Assumption 2 and properties of entropy, $\min_{\theta} \rho(\theta) > 0$ and $\max_{\theta} \rho(\theta) < \infty$. In addition to Assumptions 2 and 3, we will need the following assumption.

**Assumption 4.** $\nabla \rho(\theta)$ is Lipschitz and $L > 0$ is the smallest number such that $\|\nabla \rho(\theta) - \nabla \rho(\theta')\| \leq L \|\theta - \theta'\|$, for all $\theta, \theta' \in \Theta$.

We have the following convergence rate result for the projected gradient descent scheme (41).

**Theorem 5.** Let Assumptions 2, 3, and 4 hold. Let $D_\zeta = \max_{z,z' \in (\zeta \circ \lambda)(\Theta)} \|z - z'\|$ denote the diameter of the convex, compact set $(\zeta \circ \lambda)(\Theta)$. Define $M = \max_{\theta \in \Theta} \rho(\theta)$, $m = \min_{\theta \in \Theta} \rho(\theta)$, $K = \max\{m^2 L, M^2 m^2 L\}$, and $L_1 = \max\{L, M^2 L\}$. Then, with $\eta = 1/K$, we have, for all $t \geq 0$,

$$\rho(\theta_t) - \rho(\theta^*) \leq \frac{4M^2L^2 D^2}{t+1}.$$

(42)

**Proof.** The key idea behind the proof is to link the objective function $\rho(\theta)$ that the updates (41) are minimizing to the concave structure of the transformed problem (Q2). This will allow us to derive the bound (42) by studying related bounds for the concave objective function of (Q3). The proof proceeds as follows: first, we convert the descent scheme (41) to an equivalent ascent scheme; next, we link the objective that this ascent scheme is maximizing to the objective function of (Q3); third, we derive a sequence capturing the convergence rate of the ascent scheme; finally, we relate this sequence back to (41) to obtain (42).

In order to more easily make the connection to the concave maximization problem (Q3), we first transform (41) into an equivalent projected gradient ascent scheme. Define $q(\theta) = 1/\rho(\theta) = (k + H(d_a))/J(\theta)$, and notice that $\nabla q(\theta) = -\nabla \rho(\theta)/[\rho(\theta)]^2$. The projected gradient ascent scheme can then be written

$$\theta_{t+1} = \text{Proj}_\Theta (\theta_t - \eta \nabla q(\theta_t))$$

(a) = \text{Proj}_\Theta \left( \theta_t + \eta [\rho(\theta_t)]^2 \nabla q(\theta_t) \right)

(b) = \arg\max_{\theta} \left( q(\theta_t) + \langle \nabla q(\theta_t), \theta - \theta_t \rangle - \frac{[\rho(\theta_t)]^2}{2\eta} \|\theta - \theta_t\|^2 \right),

(c) = \arg\max_{\theta} \left( q(\theta_t) + \langle \nabla q(\theta_t), \theta - \theta_t \rangle - \frac{1}{2\eta_t} \|\theta - \theta_t\|^2 \right).

(43)
Recall that \((c)\) holds by definition of the Proj operator, and \((c)\) results from defining \(\eta_t = \eta (\rho (\theta_t))^2 = (\rho (\theta_t))^2 / K_0\).

In order to analyze the updates \((43)\) and eventually tie the analysis back to \((41)\), we next identify a family \(\{ K_c \mid c \in [m,M] \}\) of Lipschitz constants of the gradient \(\nabla q (\theta)\). By Assumption \((2)\), \(\nabla q (\theta)\) is Lipschitz and \(\| \nabla q (\theta) - \nabla q (\theta') \| \leq L_0 \| \theta - \theta' \|\), for all \(\theta, \theta' \in \Theta\), where \(L_0 = m^2 L\). Let \(K = \max \{ L_0, M^2 L_0 \} = \max \{ m^2 L, M^2 m^2 L \}\). Then, for all scalars \(c \in [m,M]\), \(\nabla q (\theta)\) satisfies \(\| \nabla q (\theta) - \nabla q (\theta') \| \leq K_c \| \theta - \theta' \|\), where \(K_c = K / c^2\) is the desired Lipschitz constant. These will be critical in the analysis to follow.

We now link the updates \((43)\) and objective function \(q (\theta)\) to \((2)\), then study a related sequence to capture the convergence rate of \((43)\). The overall goal is to study the sequence \(\alpha_t = [q (\theta^*) - q (\theta_t)] / 2 K m \ell^2 D_c^2\), for \(t \geq 0\), ultimately using properties of the sequence to show that \(q (\theta^*) - q (\theta_t) \leq 4 K_m \ell^2 D_c^2 / (1 + t)\), for all \(t \geq 0\). We will then use this inequality to obtain the desired bound \((12)\). The remainder of the proof proceeds along lines similar to that of \((35)\), but with modifications to accommodate the use of the perspective transform, the variable transformation \(\zeta\), and the non-constant stepsizes \(\eta_t = \eta (\rho (\theta_t))^2\).

Before proceeding, we make some necessary definitions. Define \(\hat{H} (\lambda) = H (d^\lambda)\), where \(H (d^\lambda)\) is the entropy of the state occupancy measure \(d^\lambda (s) = \sum_a \lambda (s, a)\) corresponding to the state-action occupancy measure \(\lambda \in D(S \times A)\). For a given \(\kappa \geq 0\), let \(P_{\kappa, \hat{H}} : \mathbb{R}^{\lambda (|S| \times |A| + 1)} \rightarrow \mathbb{R}\) denote the perspective transformation of \(\kappa + \hat{H} (\lambda)\), given by \(P_{\kappa, \hat{H}} (z) = P_{\kappa, \hat{H}} ((y, t)) = t (\kappa + \hat{H} (y / t))\). For notational convenience we henceforth drop the dependency on \(\kappa\) and simply write \(P_{\hat{H}}\).

Our next step is to make use of the concave structure of \(P_{\hat{H}}\), \(\zeta \circ \lambda\), and the associated problem \((Q_3)\) to derive properties that will aid our analysis of \(\{ \alpha_t \}_{t \in \mathbb{N}}\). We will first spend some time deriving several useful inequalities regarding \(P_{\hat{H}}\) and \(\zeta \circ \lambda\) before returning to our analysis of \(\{ \alpha_t \}_{t \in \mathbb{N}}\). Notice that \(q (\theta) = (\kappa + \hat{H} (\lambda_0)) / J (\lambda_0) = P_{\hat{H}} (\zeta (\lambda_0)) = P_{\hat{H}} (\zeta (\lambda (\theta)))\), for all \(\theta \in \Theta\). This means that we can rewrite \(\alpha_t\) as
\[
\alpha_t = \frac{P_{\hat{H}} (\zeta \circ \lambda (\theta^*)) - P_{\hat{H}} (\zeta \circ \lambda (\theta_t))}{2 K m \ell^2 D_c^2}.
\]

Recall that \(P_{\hat{H}}\) is concave over the region \(\zeta (D (S \times A))\), since \(\hat{H}\) is concave over the region \(D (S \times A)\) and the perspective transform preserves concavity. \(P_{\hat{H}}\) is thus concave over the convex compact region \((\zeta \circ \lambda) (\Theta) \subseteq \zeta (D (S \times A))\). Furthermore, since \(P_{\hat{H}} (\zeta \circ \lambda (\theta)) = q (\theta)\), we have \(\nabla P_{\hat{H}} (\zeta \circ \lambda (\theta)) = \nabla q (\theta)\), so \(\nabla P_{\hat{H}} (\zeta \circ \lambda (\theta))\) is \(K_c\)-Lipschitz, for any \(K_c = K / c^2\), \(c \in [m,M]\). This implies (see \((31)\) Lemma 1.2.3)), for any \(c \in [m,M]\), that
\[
\left| P_{\hat{H}} (\zeta \circ \lambda (\theta_t)) - P_{\hat{H}} (\zeta \circ \lambda (\theta_0)) - \langle \nabla P_{\hat{H}} (\zeta \circ \lambda (\theta_t)), \theta - \theta_t \rangle \right| \leq \frac{K_c}{2} \left\| \theta - \theta_t \right\|^2.
\]
whence, for any \(\theta \in \Theta\),
\[
P_{\hat{H}} (\zeta \circ \lambda (\theta_t)) \geq P_{\hat{H}} (\zeta \circ \lambda (\theta_0)) + \langle \nabla P_{\hat{H}} (\zeta \circ \lambda (\theta_t)), \theta - \theta_t \rangle - \frac{K_c}{2} \left\| \theta - \theta_t \right\|^2.
\]
In light of these inequalities, and using the fact that \(\eta_t = [\rho (\theta_t)]^2 / K = 1 / K \rho (\theta_t)\) by setting \(c = \rho (\theta_t)\) in the definition of \(K_c\), we have

\[
P_{\hat{H}} (\zeta \circ \lambda (\theta_{t+1})) \geq P_{\hat{H}} (\zeta \circ \lambda (\theta_t)) + \langle \nabla P_{\hat{H}} (\zeta \circ \lambda (\theta_t)), \theta_{t+1} - \theta_t \rangle - \frac{K_c}{2} \left\| \theta_{t+1} - \theta_t \right\|^2,
\]

\[
\overset{(a)}{=} \max_{\theta \in \Theta} \left( P_{\hat{H}} (\zeta \circ \lambda (\theta_t)) + \langle \nabla P_{\hat{H}} (\zeta \circ \lambda (\theta_t)), \theta - \theta_t \rangle - \frac{K_c}{2} \left\| \theta - \theta_t \right\|^2 \right)
\]

\[
\overset{(b)}{=} \max_{\theta \in \Theta} \left( P_{\hat{H}} (\zeta \circ \lambda (\theta)) - K_c \left\| \theta - \theta_t \right\|^2 \right)
\]

\[
\overset{(c)}{=} \max_{\theta \in \Theta} \left( P_{\hat{H}} (\zeta \circ \lambda (\theta)) - K_m \left\| \theta - \theta_t \right\|^2 \right)
\]

\[
\overset{(d)}{=} \max_{\alpha \in [0,1]} \left\{ P_{\hat{H}} (\zeta \circ \lambda (\theta_0)) - K_m \left\| \theta_0 - \theta_1 \right\|^2 \right\} \overset{\theta_0 = k (\alpha (\zeta \circ \lambda (\theta^*)) + (1 - \alpha) (\zeta \circ \lambda (\theta_t)))}{,}
\]

\[
(46)
\]
where (a) follows from the optimality of the update (43), (b) holds by (45), (c) follows from the fact that $K_{P(\theta)} \leq K_m$, the (d) follows by the convexity of $(\zeta \circ \lambda)(\Theta)$. Now notice that

$$P_{\tilde{H}}((\zeta \circ \lambda)(\theta_\alpha)) = P_{\tilde{H}}\left((\zeta \circ \lambda)(k(\alpha(\zeta \circ \lambda)(\theta^*) + (1 - \alpha)(\zeta \circ \lambda)(\theta_t))\right)$$

$$= P_{\tilde{H}}(\alpha(\zeta \circ \lambda)(\theta^*) + (1 - \alpha)(\zeta \circ \lambda)(\theta_t))$$

$$\geq \alpha P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) + (1 - \alpha)P_{\tilde{H}}((\zeta \circ \lambda)(\theta_t)), \quad (47)$$

where the first equality holds by the definition of $\theta_\alpha$ given in (46), the second follows from the fact that $k((\zeta \circ \lambda)(\theta)) = \theta$, for any $\theta \in \Theta$, and the final inequality is yielded by the concavity of $P_{\tilde{H}}$ over $(\zeta \circ \lambda)(\Theta)$. Furthermore,

$$\|\theta_\alpha - \theta_t\|^2 = \|k(\alpha(\zeta \circ \lambda)(\theta^*) + (1 - \alpha)(\zeta \circ \lambda)(\theta_t)) - k((\zeta \circ \lambda)(\theta_t))\|^2$$

$$\leq \ell^2 \|\alpha(\zeta \circ \lambda)(\theta^*) + (1 - \alpha)(\zeta \circ \lambda)(\theta_t) - (\zeta \circ \lambda)(\theta_t)\|^2$$

$$\leq \alpha^2 \ell^2 \|\zeta^\circ(\lambda)(\theta^*) - (\zeta \circ \lambda)(\theta_t)\|^2$$

$$\leq \alpha^2 \ell^2 D_\zeta^2, \quad (48)$$

where (a) holds by the definition of $\theta_\alpha$ and the fact that $k((\zeta \circ \lambda)(\theta)) = \theta$, (b) follows since $k$ is $\ell$-Lipschitz, and (c) results from the definition of $D_\zeta$ given in the statement of the theorem. Now, the inequalities (46), (47), and (48) combine to yield

$$P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - P_{\tilde{H}}((\zeta \circ \lambda)(\theta_{t+1}))$$

$$\leq \min_{\alpha \in [0,1]} \{P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - P_{\tilde{H}}((\zeta \circ \lambda)(\theta_\alpha)) + K_m \|\theta_\alpha - \theta_t\|^2\}$$

$$\leq \min_{\alpha \in [0,1]} \left(P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - \alpha P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - (1 - \alpha)P_{\tilde{H}}((\zeta \circ \lambda)(\theta_t)) + K_m \alpha^2 \ell^2 D_\zeta^2\right)$$

$$= \min_{\alpha \in [0,1]} \left(1 - \alpha\right)\left(P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - P_{\tilde{H}}((\zeta \circ \lambda)(\theta_t)) + K_m \alpha^2 \ell^2 D_\zeta^2\right), \quad (49)$$

where inequality (d) results from multiplying both sides of (46) by $-1$ and adding $P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*))$, and (47) and (48) together yield (e). Now that we have the important inequalities (46), (47), (48), and (49) in hand, we return to our analysis of the sequence $\{\alpha_t\}_{t \in \mathbb{N}}$ defined in (44). Our goal is to derive a useful upper bound for the sequence, then use this bound to prove that, when the iterates $\{\theta_t\}$ are generated by (43), $q(\theta_t)$ converges to $q(\theta^*)$ at the certain rate. We will then use these results to prove (12).

To begin our analysis of the sequence, we use (49) to derive a useful recursive inequality for $\{\alpha_t\}_{t \in \mathbb{N}}$. Notice that $\alpha_t \geq 0$, for all $t \geq 0$. Now, assume that $\alpha_0 \geq 1$. This implies that $P_{\tilde{H}}((\zeta \circ \lambda)(\theta^*)) - P_{\tilde{H}}((\zeta \circ \lambda)(\theta_0)) \geq 2K_m \ell^2 D_\zeta^2$, so the minimum in (49) is attained when $\alpha = 1$. But then $\alpha_1 \leq 1/2$. Since this argument is independent of the choice of $t$, we can assume without loss of generality that $\alpha_t \leq 1$, for all $t \geq 0$, by simply discarding $\alpha_0$ if it is greater than 1.

We next show that $\alpha_{t+1} \leq \alpha_t$, for all $t \geq 0$. Since $\alpha_t \leq 1$, $\alpha_t$ is always the minimizer of the right-hand side of (49), which can be seen by setting the derivative with respect to $\alpha$ equal to 0 and solving for $\alpha$. 

26
Substituting $\alpha_t$ into (49), we see that
\[
P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_{t+1})) \\
\leq \left(1 - \frac{P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))}{2K_m\ell^2D_\zeta^2}\right)\left(P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))\right) \\
+ \left(\frac{P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))}{2K_m\ell^2D_\zeta^2}\right)^2K_m\ell^2D_\zeta^2 \\
= \left(1 - \frac{P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))}{2K_m\ell^2D_\zeta^2}\right)\left(P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))\right) \\
+ \frac{P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))}{4K_m\ell^2D_\zeta^2}(P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))) \\
= \left(1 - \frac{P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))}{4K_m\ell^2D_\zeta^2}\right)(P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))),
\]
where (a) results by noticing that one of the $K_m\ell^2D_\zeta^2$ terms cancels and (b) can be obtained by factoring out the $P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t))$ term and simplifying. Dividing both sides of (50) by $2K_m\ell^2D_\zeta^2$ shows that $\alpha_{t+1} \leq \alpha_t$.

Now, dividing both sides of (50) by $4K_m\ell^2D_\zeta^2$ yields the recursive inequality
\[
\frac{\alpha_{t+1}}{2} \leq \left(1 - \frac{\alpha_t}{2}\right)\frac{\alpha_t}{2},
\]
which implies
\[
\frac{2}{\alpha_{t+1}} \geq \frac{1}{1 - \frac{\alpha_t}{2}} = \frac{2\left(1 - \frac{\alpha_t}{2}\right) + \alpha_t}{(1 - \frac{\alpha_t}{2})\alpha_t} = \frac{2}{\alpha_t} + \frac{1}{1 - \frac{\alpha_t}{2}} \geq \frac{2}{\alpha_t} + 1 \geq \frac{2}{\alpha_0} + t.
\]
Since $\alpha_0 \leq 1$, this gives us that
\[
\frac{\alpha_{t+1}}{2} \leq \frac{1}{t + \frac{2}{\alpha_0}} \leq \frac{1}{t + 2}.
\]
Multiplying both sides of (51) by $4K_m\ell^2D_\zeta^2$ finally yields, for all $t \geq 0$,
\[
q(\theta^*) - q(\theta_t) = P_H((\zeta \circ \lambda)(\theta^*)) - P_H((\zeta \circ \lambda)(\theta_t)) \leq \frac{4K_m\ell^2D_\zeta^2}{t + 1}.
\]
We have thus finally obtained our convergence rate result for the projected gradient ascent scheme (43). We now finish the proof by using this result to derive the corresponding rate for the OIR projected gradient descent scheme (44). Since $q(\theta) = 1/\rho(\theta)$ and $K_m = K/m^2$, (52) implies that
\[
\rho(\theta_t) - \rho(\theta^*) \leq \rho(\theta_t) - \rho(\theta^*) = \frac{1}{\rho(\theta^*)} - \frac{1}{\rho(\theta_t)} = q(\theta^*) - q(\theta_t) \leq \frac{4K_m\ell^2D_\zeta^2}{t + 1} = \frac{4K^2\ell^2D_\zeta^2}{m^2(t + 1)}.
\]
Remembering that $K = \max\{m^2L, M^2m^2L\}$ and letting $L_1 = \max\{L, M^2L\}$, we have
\[
\rho(\theta_t) - \rho(\theta^*) \leq \frac{M^2 4K^2\ell^2D_\zeta^2}{m^2(t + 1)} = \frac{4M^2L_1\ell^2D_\zeta^2}{t + 1},
\]
which completes the proof. \(\square\)

Coupled with Theorem 4, this result provides a non-asymptotic convergence rate to global optimality for algorithms solving the OIR minimization problem (31) when exact gradient evaluations are available. When compared with the corresponding result in [45], to which it is closely related, the bound of Theorem 5 contains an interesting dependence on the value of the user-specified $\kappa$, the policy class $\{\pi_\theta\}_{\theta \in \Theta}$, and the underlying MDP dynamics; this dependence is discussed in the following remark.
Remark 3. The presence of $M = \max_{\theta \in \Theta} \rho(\theta) = \max_{\theta}[J(\theta)/(\kappa + H(d_0))]$ in the bound \cite{42} suggests that the convergence rate depends on the value of $\kappa$ as well as the minimal possible value of $H(d_0)$ over the set $\Theta$. To see why, let $C = \max_{\theta \in \Theta} J(\theta)$ and notice that

$$M \leq \max_{\theta \in \Theta} \frac{C}{\kappa + H(d_0)} = \frac{C}{\kappa + \min_{\theta \in \Theta} H(d_0)}. \quad (53)$$

On the one hand, when $\kappa + \min_{\theta \in \Theta} H(d_0)$ is very large, the value of $M$ will be close to 0, yielding a tighter bound in \cite{42}. This trivially occurs when the user-specified constant $\kappa$ is chosen to be large, since the overall objective function $\rho(\theta)$ can be forced artificially close to 0. More interestingly, if the dynamics of the underlying MDP and/or the policy class $\{\pi_\theta\}_{\theta \in \Theta}$ are such that all possible policies induce relatively high-entropy state occupancy measures, then the value of $\min_{\theta \in \Theta} H(d_0)$ will be larger and the value of $M$ smaller. This suggests that it may be easier to optimize the OIR over MDPs and/or policy classes that tend to be “more ergodic”. When $\kappa + \min_{\theta \in \Theta} H(d_0)$ is close to 0, on the other hand, the value of $M$ may be very large, resulting in a looser bound in \cite{42}. This occurs when both the constant $\kappa$ is chosen to be very small and the MDP dynamics and/or the policy class $\{\pi_\theta\}_{\theta \in \Theta}$ have the potential to induce very low-entropy occupancy measures. This highlights the usefulness of the constant $\kappa$, as it can be used to smooth the objective function $\rho(\theta)$ and thereby lead to stabler convergence when optimizing the OIR over MDPs and policy classes that tend to be “less ergodic”.

We now have non-asymptotic convergence guarantees for policy gradient algorithms solving the OIR problem when gradients can be evaluated exactly. Nonetheless, it would be nice to have convergence guarantees when only noisy gradient estimates are available, as is typically the case in practice. To this end, we next prove almost sure convergence of the IDAC algorithm developed in Section 6.2.

7.3 Actor-Critic Convergence

We conclude this section by proving almost sure (a.s.) convergence of IDAC to a neighborhood of a stationary point of the optimization problem $\min_{\theta \in \Theta} \rho(\theta)$. As discussed in Section 7.1 this implies that, thanks to the hidden quasiconcavity of the OIR optimization problem, IDAC converges a.s. to a neighborhood of a global optimum. This result is much stronger than existing asymptotic results for actor-critic schemes, which typically only guarantee convergence to a neighborhood of a local optimum or even a mere saddle point \cite{10, 49, 2}.

We analyze the algorithm as given in equations (24)-(30) under the assumption that $\tau_t = \alpha_t$, for all $t \geq 0$, and with the addition of a projection operation to equation (30):

$$\theta_{t+1} = \Gamma \left( \theta_t - \beta_t \delta_t \left( \kappa + \mu_t^H \right) - \mu_t^H \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \right), \quad (54)$$

where $\Gamma : \mathbb{R}^d \to \Theta$ maps any parameter $\theta \in \mathbb{R}^d$ back onto the compact set $\Theta \subset \mathbb{R}^d$ of permissible policy parameters. This projection, which is common in the actor-critic and broader two-timescale stochastic approximation literatures (see, e.g., \cite{23, 11, 10}) is for purposes of theoretical analysis, and is typically not needed in practice. For ease of exposition, we also assume access to the oracle DENSITYESTIMATOR discussed in Section 6.1 which returns the occupancy measure $d_0 = DENSITYESTIMATOR(\theta)$ when provided with input policy parameter $\theta \in \Theta$. In addition to Assumption 2 we require the following.

Assumption 5. The stepsize sequences $\{\alpha_t\}$ and $\{\beta_t\}$ satisfy $\sum_t \alpha_t = \sum_t \beta_t = \infty$, $\sum_t \alpha_t^2 + \beta_t^2 < \infty$, and $\lim_t \frac{\beta_t}{\alpha_t} = 0$.

Assumption 6. The value function approximators $\psi_\omega$ are linear, i.e. $\psi_\omega(s) = \omega^\top \phi(s)$, where $\phi(s) = [\phi_1(s) \cdots \phi_K(s)]^\top \in \mathbb{R}^K$ is the feature vector associated with $s \in \mathcal{S}$. The feature vectors $\phi(s)$ are uniformly bounded for any $s \in \mathcal{S}$, and the feature matrix $\Phi = [\phi(s)]_{s \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S} \times K}$ has full column rank. For any $u \in \mathbb{R}^K$, $\Phi u \neq 1$, where $1$ is the vector of all ones.

Assumptions 2, 5 and 6 are all standard in two-timescale convergence analyses for actor-critic algorithms \cite{10}. Assumption 5 is common to almost all two-timescale stochastic approximation schemes \cite{11}, and it...
guarantees the appropriate separation of timescales between the actor and critic portions of the algorithm. Assumption 6 is needed to guarantee convergence of the TD(0) updates used for the critic in Lemma 5 [44, 10]. Assumption 2 is needed to ensure boundedness of the actor parameters, a critical condition for stochastic approximation schemes to hold [23]. Assumption 2 is also needed to ensure the existence of the gradients in Lemma 6 and guarantee that the ODE considered in Theorem 6 is well-posed.

We now proceed with the results. Our analysis leverages the average-reward actor-critic results in [10] as well as the results for ratio optimization actor-critic in [39]. For a given policy parameter \( \theta \), let \( D_\theta = \text{diag}(d_\theta) \in \mathbb{R}^{|S|} \) denote the matrix with the elements of \( d_\theta \) along the diagonal and zeros everywhere else. Define the state cost vector for the average-cost MDP \( c_s \) and let \( \text{diag}(c_s) \).

Under Assumption 6, given a fixed policy parameter \( \theta \), let \( \nu_0(s) \in \mathbb{R}^{|S|} \) denote the state transition probability matrix under policy \( \pi_0 \), i.e., \( \nu_0(s'|s) = \sum_{a \in A} \pi_0(a|s)p(s'|s,a) \), for any \( s, s' \in S \). We first show convergence of the critics.

**Lemma 5.** Under Assumption 6, given a fixed policy parameter \( \theta \in \Theta \), the recursive updates (24)–(29) converge as follows: \( \lim_{t \to \infty} \mu_t^J = J(\theta) \) a.s., \( \lim_{t \to \infty} \mu_t^H = H(\theta) \) a.s., \( \lim_{t \to \infty} \omega_t^J = \omega_0^J \) a.s., and \( \lim_{t \to \infty} \omega_t^H = \omega_0^H \) a.s., where \( \omega_0^J \) and \( \omega_0^H \) are, respectively, the unique solutions to

\[
\Phi^\top D_\theta [c_0 - J(\theta) \cdot 1 + P_\theta (\Phi \omega^J) - \Phi \omega^J] = 0, \tag{55}
\]

\[
\Phi^\top D_\theta [\tau_0 - H(\nu_0) \cdot 1 + P_\theta (\Phi \omega^H) - \Phi \omega^H] = 0. \tag{56}
\]

**Proof.** Since the policy \( \pi_0 \) is held fixed and the shadow MDP reward \( -\log d_\theta(s) \) can be exactly evaluated, for any \( s \in S \), the proof of [10] Lemma 4 can be applied separately to the average-cost recursions (24), (26), (28) and the shadow MDP recursions (27), (29), (29) to obtain the result.

As in [10] Lemma 5, this result shows that the sequences \( \{\omega_t^J\} \) and \( \{\omega_t^H\} \) converge a.s. to the limit points \( \omega_0^J \) and \( \omega_0^H \) of the TD(0) algorithm with linear function approximation for their respective MDPs. Due to the linearity of the function approximation, when used in the policy update step the value function estimates \( v_0^J = \Phi \omega_0^J \) and \( v_0^H = \Phi \omega_0^H \) may result in biased gradient estimates. Similar to the bias characterization given in [10] Lemma 4, this bias can be characterized as follows.

**Lemma 6.** Fix \( \theta \in \Theta \). Let

\[
\delta_t^J = c(s_t, a_t) - J(\theta) + \phi(s_{t+1})^\top \omega_0^J - \phi(s_t)^\top \omega_0^J,
\]

\[
\delta_t^H = -\log d_\theta(s_t) - H(\nu_0) + \phi(s_{t+1})^\top \omega_0^H - \phi(s_t)^\top \omega_0^H,
\]

denote the stationary estimates of the TD-errors at time \( t \). Let

\[
\tau_t^J = \mathbb{E}_{\pi_0}[c(s,a) - J(\theta) + \phi(s')^\top \omega_0^J],
\]

\[
\tau_t^H = \mathbb{E}_{\pi_0}[-\log d_\theta(s) - H(\nu_0) + \phi(s')^\top \omega_0^H],
\]

and let

\[
\epsilon_t^J = \sum_{s \in S} d_\theta(s) [\nabla_\theta \tau_t^J(s) - \nabla_\theta \phi(s)^\top \omega_0^J]
\]

\[
\epsilon_t^H = \sum_{s \in S} d_\theta(s) [\nabla_\theta \tau_t^H(s) - \nabla_\theta \phi(s)^\top \omega_0^H].
\]

We then have that

\[
\mathbb{E}_{\pi_\theta} \left[ \frac{\delta_t^J \kappa + H(\nu_0)}{[\kappa + H(\nu_0)]^2} \nabla \log \pi_\theta(a_t|s_t) \right] = \nabla \rho(\theta) + \frac{\epsilon_t^J \kappa + H(\nu_0)}{[\kappa + H(\nu_0)]^2}.
\]

**Proof.** By [10] Lemma 4 and Theorem 2,

\[
\mathbb{E}_{\pi_\theta} \left[ \delta_t^J \nabla \log \pi_\theta(a_t|s_t) \right] = \nabla J(\theta) + \epsilon_t^J,
\]

\[
\mathbb{E}_{\pi_\theta} \left[ \delta_t^H \nabla \log \pi_\theta(a_t|s_t) \right] = \nabla H(\nu_0) + \epsilon_t^H.
\]
This implies that
\[
\mathbb{E}_{\pi_\theta} \left[ \frac{\delta_t^{\theta,J} [\kappa + H(d_0)] - J(\theta)\delta_t^H}{[\kappa + H(d_0)]^2} \nabla \log \pi_\theta(a_t|s_t) \right]
= \frac{[\kappa + H(d_0)]}{[\kappa + H(d_0)]^2} \mathbb{E}_{\pi_\theta} \left[ \delta_t^{\theta,J} \nabla \log \pi_\theta(a_t|s_t) - J(\theta) \mathbb{E}_{\pi_\theta} \left[ \delta_t^H \nabla \log \pi_\theta(a_t|s_t) \right] \right]
= \frac{[\kappa + H(d_0)]}{[\kappa + H(d_0)]^2} \left( \nabla J(\theta) + \epsilon_{\theta}^J - J(\theta) \epsilon_{\theta}^H \right)
= \nabla \rho(\theta) + \epsilon_{\theta}^J \frac{[\kappa + H(d_0)] - J(\theta)\epsilon_{\theta}^H}{[\kappa + H(d_0)]^2},
\]
which completes the proof.

Now we are ready to establish the convergence of the actor step, and thus the actor-critic algorithm. Given any continuous function \( f : \Theta \to \mathbb{R}^d \), define the function \( \hat{\Gamma}(\cdot) \) using the projection operator \( \Gamma \) to be
\[
\hat{\Gamma}(f(\theta)) = \lim_{\eta \to 0^+} [\Gamma(\theta + \eta \cdot f(\theta)) - \theta] / \eta.
\]
Define
\[
\epsilon_{\theta} = \epsilon_{\theta}^J \frac{[\kappa + H(d_0)] - J(\theta)\epsilon_{\theta}^H}{[\kappa + H(d_0)]^2},
\] and consider the ordinary differential equations (ODEs)
\[
\dot{\theta} = \hat{\Gamma}(\nabla \rho(\theta)),
\]
\[
\dot{\theta} = \hat{\Gamma}(\nabla \rho(\theta) + \epsilon_{\theta}).
\] Notice that, by the definition of \( \hat{\Gamma} \), the right-hand side of (58) is simply \( \Gamma(\nabla \rho(\theta)) \) when there exists \( \eta_0 > 0 \) such that \( \theta + \eta \nabla \rho(\theta) \in \Theta \), for all \( \eta < \eta_0 \). When such an \( \eta_0 \) does not exist, \( \Gamma(\nabla \rho(\theta)) \) can be interpreted as the projected ODE \( \dot{\theta} = \nabla \rho(\theta) + z(\theta) \), where \( z(\theta) \) is the minimal force necessary to project \( \theta \) back onto \( \Theta \). Similar statements hold for (59). For further discussion of the definition of \( \hat{\Gamma} \) and related results, see [23, p. 191]. For the projected ODE interpretation, see [23 §4.3].

We now present the main result of this subsection, which establishes convergence of the actor-critic algorithm given by equations (24)-(29) and (54).

**Theorem 6.** Let \( Z, \bar{Y} \) denote the sets of asymptotically stable equilibria of the ODEs (58), (59), respectively. Given any \( \epsilon > 0 \), define \( \bar{Z}^\epsilon = \{ z | \inf_{z', \in Z} \| z - z' \| \leq \epsilon \} \). For any \( \theta \in \Theta \), let \( \epsilon_{\theta} \) be defined as in (57). Under Assumptions 2 and 3, given any \( \epsilon > 0 \), there exists \( \delta > 0 \) such that, for \( \{\theta_t\} \) obtained from the recursive scheme (24)-(29) and (54), if \( \sup_{t} \| \epsilon_{\theta_t} \| < \delta \), then \( \theta_t \to \bar{Z}^\epsilon \) a.s. as \( t \to \infty \).

**Proof.** The main idea behind the proof is to show that the update scheme (54) converges to a neighborhood of a stationary point by showing that it eventually tracks the ODE (58). To do this, we first rewrite the update in (54) as the sum of the true gradient direction and noise terms. We next argue that these noise terms are asymptotically negligible. We finally show that this noisy update scheme asymptotically tracks the biased ODE (59), which means that it converges to a neighborhood of the stationary points of (58). The proof largely follows that of [18 Theorem 1], but with key modifications to accommodate complications arising from the fact that the objective to be minimized is a ratio; specifically, we ensure that: (i) the resulting noise terms are indeed asymptotically negligible, and (ii) the Lipschitz properties of the gradient \( \nabla \rho(\theta) \) necessary for the ODE analysis are satisfied.

Let \( \mathcal{F}_t = \sigma(\theta_k, k \leq t) \) denote the \( \sigma \)-algebra generated by the \( \theta \)-iterates up to time \( t \). Define
\[
\delta_t = \frac{\delta_t^J [\kappa + \mu_t^H] - \mu_t^J}{[\kappa + \mu_t^H]^2}, \quad \delta_t^H = \frac{\delta_t^{H,\theta} [\kappa + H(d_0)] - J(\theta)\epsilon_{\theta}^H}{[\kappa + H(d_0)]^2}.
\]
In addition, define the noise terms
\[
M_{t}^{(1)} = \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) - \mathbb{E}_{\pi_{\theta_t}} [\delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t], \\
= \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) - \mathbb{E} [\delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t], \\
M_{t}^{(2)} = \mathbb{E}_{\pi_{\theta_t}} \left[ (\delta_t - \delta_t^\theta_t) \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t \right], \\
= \mathbb{E} \left[ (\delta_t - \delta_t^\theta_t) \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t \right],
\]
as well as the function
\[
h(\theta_t) = \mathbb{E}_{\pi_{\theta_t}} \left[ \delta_t^\theta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t \right] \\
= \mathbb{E}_{\pi_{\theta_t}} \left[ \delta_t^\theta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \right],
\]
which is the gradient expression from Lemma\ref{lemma:gradient}. Note that simultaneously taking an expectation with respect to \(\pi_{\theta_t}\) and conditioning on \(\mathcal{F}_t\) is redundant, so we can suppress one or the other in our notation without altering the meaning. In what follows we will typically condition on \(\mathcal{F}_t\) when we discussing the stochastic processes \(\{M_{t}^{(1)}\}, \{M_{t}^{(2)}\}\), whereas we will take expectations with respect to \(\pi_{\theta_t}\) when discussing the function \(h\).

We can now rewrite the projected actor update \([54]\) as
\[
\theta_{t+1} = \Gamma \left( \theta_t - \beta_t \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \right) \\
= \Gamma \left( \theta_t - \beta_t \left(h(\theta_t) + M_{t}^{(1)} + M_{t}^{(2)}\right) \right).
\]

We show that this update scheme asymptotically tracks the ODE \([59]\) a.s. by demonstrating that the noise terms \(\{M_{t}^{(1)}\}\) form an a.s. bounded martingale difference sequence, that the terms \(\{M_{t}^{(2)}\}\) are asymptotically negligible, and that \(h\) is Lipschitz and thus the ODE is well-posed.

First notice that, since \(\delta_t \to \delta_t^\theta_t\) a.s. by Lemma\ref{lemma:gradient}, we have that \(M_{t}^{(2)} \to 0\) a.s., so the noise terms \(\{M_{t}^{(1)}\}\) are indeed asymptotically negligible. Next, recall the tower property of conditional expectations, which states that, for any \(\mathcal{F}\)-measurable random variable \(X\) and any sub-\(\sigma\)-algebras \(G \subseteq H \subseteq \mathcal{F}\), we have
\[
\mathbb{E} \left[ \mathbb{E} [X \mid G] \mid H \right] = \mathbb{E} \left[ \mathbb{E} [X \mid H] \mid G \right] = \mathbb{E} [X \mid G].
\]
Since \(\mathcal{F}_t \subseteq \mathcal{F}_{t+1}\), for all \(t \geq 0\), this implies that, for all \(t \geq 0\),
\[
\mathbb{E} \left[ M_{t+1}^{(1)} \mid \mathcal{F}_t \right] = \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_t \right] - \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_{t+1} \right] \\
= \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_t \right] - \mathbb{E} \left[ \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_{t+1} \right] \mid \mathcal{F}_t \right] \\
= \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_t \right] - \mathbb{E} \left[ \delta_{t+1} \nabla \log \pi_{\theta_{t+1}}(a_{t+1}|s_{t+1}) \mid \mathcal{F}_t \right] \\
= 0,
\]
so \(\{M_{t}^{(1)}\}\) is an \(\mathcal{F}\)-martingale difference sequence, where \(\mathcal{F}\) is the filtration \(\mathcal{F} = \{\mathcal{F}_t\}\). To see that \(M_{t}^{(1)}\) is a.s. bounded, first notice that \(\mu_0^H > 0\), and \(0 < \inf_t d_{\theta_t}(s_t) \leq \sup_t d_{\theta_t}(s_t) \leq 1\), so by \([23]\) \(\{\mu_t^H\}\) is uniformly bounded both above and below away from zero. A similar argument applies to \(\{\mu_t^\theta\}\), and, coupled with the a.s. boundedness of \(\{M_t^\theta\}\) and \(\{M_t^H\}\), this implies that \(\{\delta_t\}\) and thus \(\{M_{t}^{(1)}\}\) are a.s. bounded. As discussed in [11, §§2.1-2.2], the facts that \(\{M_{t}^{(1)}\}\) is a.s. bounded martingale difference noise and \(\{M_{t}^{(2)}\}\) is asymptotically negligible ensure that, so long as the right-hand side of the ODE \([59]\) is Lipschitz, the iterates generated by \([54]\) will asymptotically track it.

To see that \(h\) is Lipschitz in \(\theta\), first rewrite
\[
h(\theta_t) = \frac{1}{[\kappa + H(d_{\theta_t})]^2} \left[ [\kappa + H(d_{\theta_t})] \mathbb{E}_{\pi_{\theta_t}} \left[ \delta^\theta_t \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \right] \right] - J(\theta_t) \mathbb{E}_{\pi_{\theta_t}} \left[ \delta^\theta_t \delta_t \nabla \log \pi_{\theta_t}(a_t|s_t) \right],
\]
We verify that each of the component terms in this expression is Lipschitz and bounded. Recall that a function is Lipschitz if it is continuously differentiable with bounded derivatives. First, as discussed in the proof of [10]
Lemma 5], \( J(\theta), d_\theta(s), \nabla \pi_\theta(a|s), \) and \( \Phi \omega_\theta^H \), are all Lipschitz and bounded, for all \( s \in \mathcal{S}, \alpha \in \mathcal{A} \). Together, these imply that the terms \( J(\theta) \) and \( \mathbb{E}_{\pi_\theta} \left[ \delta_t^{H,\theta} \left( - \log \pi_\theta(a_t|s_t) \right) \right] \) are Lipschitz and bounded on \( \Theta \). This means that the only remaining terms we need to inspect are \( \kappa + H(d_\theta), 1/|\kappa + H(d_\theta)|^2 \), and \( \mathbb{E}_{\pi_\theta} \left[ \delta_t^{H,\theta} \left( - \log \pi_\theta(a_t|s_t) \right) \right] \).

Theorem 2 implies \( \nabla H(d_\theta) \) is continuous and bounded. To see this, notice that
\[
\nabla H(d_\theta) = \mathbb{E}_{\pi_\theta} \left[ \left( - \log d_\theta(s) - H(d_\theta) \right) \nabla \log \pi_\theta(a|s) \right] \\
= \sum_s d_\theta(s) \sum_a \pi_\theta(a|s) \left[ ( - \log d_\theta(s) - H(d_\theta)) \nabla \log \pi_\theta(a|s) \right] \\
= \sum_s d_\theta(s) \sum_a \nabla \pi_\theta(a|s) \left[ ( - \log d_\theta(s) - H(d_\theta)) \right].
\]

By the ergodicity condition of Assumption 2 we have that \( d_\theta(s) > 0 \), for all \( s \in \mathcal{S} \), which means that the \( - \log d_\theta(s) \) is always defined. Since \( d_\theta \) is continuous, we furthermore have that \( - \log d_\theta(s) \) and \( H(d_\theta) \) are both continuous. The gradient \( \nabla \pi_\theta(a|s) \) is continuous by Assumption 2. Finally, since \( \Theta \) is a compact set, we know that \( d_\theta(s), \nabla \pi_\theta(a|s), - \log d_\theta(s), \) and \( H(d_\theta) \) remain bounded, implying that \( \nabla H(d_\theta) \) is continuous and bounded, since it is formed by taking products and sums of continuous, bounded functions. \( H(d_\theta) \) is thus Lipschitz and bounded, as is the term \( \kappa + H(d_\theta) \), for any constant \( \kappa \geq 0 \). Furthermore, since \( d_\theta(s) > 0 \), for all \( s \in \mathcal{S} \), and since \( \Theta \) is compact, there exists some constant \( B \) such that \( \inf_{\theta \in \Theta} H(d_\theta) = B > 0 \). This means that \( 1/|\kappa + H(d_\theta)|^2 \leq 1/|\kappa + B|^2 \), for all \( \theta \in \Theta \). The term \( 1/|\kappa + H(d_\theta)|^2 \) is therefore Lipschitz and bounded, as well.

We now turn to the last remaining term. Notice that
\[
\mathbb{E}_{\pi_\theta} \left[ \delta_t^{H,\theta} \left( - \log \pi_\theta(a_t|s_t) \right) \right] = \mathbb{E}_{\pi_\theta} \left[ \left( - \log d_\theta(s_t) - H(d_\theta) + \phi(s_t, \omega_H^\theta, \omega_\theta^H) \right) \nabla \log \pi_\theta(a_t|s_t) \right] \\
= \sum_s d_\theta(s) \nabla \pi_\theta(a|s) \left[ - \log d_\theta(s) - H(d_\theta) + \phi(s) \omega_H^\theta + \sum_{s'} p(s'|s,a) \phi(s') \omega_\theta^H \right].
\]

As discussed above, \( d_\theta(s), \pi_\theta(a|s), - \log d_\theta(s), \) and \( H(d_\theta) \) are all continuously differentiable with bounded derivatives on \( \Theta \). Furthermore, given that \( H(d_\theta) \) is Lipschitz and bounded both above and away from zero, \( \Phi \omega_\theta^H \) is Lipschitz and bounded for reasons analogous to those for \( \Phi \omega_\theta^H \). Expression (60) is thus Lipschitz and bounded. The function \( h \) is thus Lipschitz as well, since it is formed by taking products and sums of Lipschitz, bounded functions.

The ODE (59) is therefore well-posed, and its equilibrium set \( \mathcal{Z} \) is well-defined. A similar argument to the one just presented can be used to show that (58) with equilibrium set \( \mathcal{Y} \) is also well-posed. The remainder of the arguments in the proof of (10) Lemma 5 now apply to prove that \( \theta_t \to \mathcal{Y} \) a.s. as \( t \to \infty \), and that, as \( \sup_{\theta} \| \varepsilon_\theta \| \to 0 \), the trajectories of (59) converge to those of (58). In particular, this implies that, for a given \( \varepsilon > 0 \), there exists a \( \delta > 0 \) such that, if \( \sup_{\theta} \| \varepsilon_\theta \| < \delta \), then \( \theta_t \to \mathcal{Z}^\varepsilon \) a.s. as \( t \to \infty \).

Combined with Theorem 3 Theorem 6 establishes almost sure convergence of IDAC to a neighborhood of a global optimum of the OIR minimization problem (31) when linear function approximation is used for the critic. Note that if the linear approximation and features are expressive enough, then the neighborhood radius \( \varepsilon \) will be small or even zero.

8 Experiments

As discussed in the introduction, OIR methods balance minimizing cost with maintaining a healthy skepticism about their environment, resisting the temptation to become overconfident based on past experience. This keeps the agent vigilant and on guard against potential black swan events—very low-probability events that can have powerful and destabilizing impacts when they do occur. A colorful example of a black swan event taken from [22] is as follows. Imagine a turkey destined to be butchered and cooked for its farmer’s Thanksgiving Day meal. During the months in the run-up to the holiday, the turkey is well-fed and cared-for by the kindly farmer. Based on its experience, the turkey’s confidence in its safety and continued well-being
grows. On the day before Thanksgiving, when the risk to its life is highest, the turkey’s confidence is ironically also highest. Due to its lack of skepticism and its overconfidence in past experience, the turkey experiences a lethal black swan event and the farmer’s Thanksgiving day meal goes as planned. Casting the turkey’s trajectory in reinforcement learning terms, it prematurely converged to a spurious, suboptimal policy by placing too much faith in its insufficiently informative past experience.

The first set of experimental results presented in this section demonstrates that, in uncertain environments, OIR policy gradient methods avoid spurious behavior, while state-of-the-art methods can become overconfident and settle into turkey-like suboptimality. In particular, these experiments demonstrate that, in settings where the reward signal is sparse, using policy gradient methods to optimize the OIR can lead to improved performance when compared with vanilla RL methods. To show this, we ran two different sets of experiments: first, we compared tabular versions of our OIR methods with vanilla actor-critic (AC) on three gridworld environments; second, we compared a neural network version of IDAC with the Stable Baselines 3 implementation of A2C, DQN, and PPO on a more complex gridworld. In all cases, the vanilla methods prematurely converge to suboptimal policies, whereas the OIR-based methods solve the problem.

The second set of experimental results presented in this section illustrates and validates the theoretical results developed in Sections 5 and 7. Specifically, the experiments show that, in the tabular setting, IDAC converges to the optimal occupancy information ratio obtained by solving the corresponding concave program. This validates the theory developed in this paper by showing that the algorithms developed in Section 6 indeed converge – as guaranteed by the results of Section 7 – to the optima of the concave program reformulation detailed in Section 5.

8.1 Improved Performance in Sparse-Reward Settings

To demonstrate the potential advantages of OIR policy gradient methods over vanilla methods in sparse-reward settings, we conducted two different sets of experiments on gridworld environments of varying complexity. In the first set of experiments, we compared tabular implementations of IDAC and vanilla AC on three relatively small gridworlds. For the second set of experiments we compared a neural network version of IDAC with the A2C, DQN, and PPO algorithms on a larger, more complex gridworld. All the environments that we considered emit sparse reward signals in the sense that the majority of costs convey no information about the central task of finding the goal state. Specific details regarding the costs are discussed below. On all four gridworlds, OIR policy gradient methods outperform the vanilla RL methods that we tested. We interpret this as illustrating that algorithms minimizing the OIR fall back to maximizing the useful alternative objective $H(d_\theta)$, representing coverage of the state space, when reward signals are sparse. This contrasts with vanilla methods, which focus on a single objective, $J(\theta)$, and can consequently converge in the sparse-reward setting to suboptimal policies before the state space has been sufficiently explored.

[https://stable-baselines3.readthedocs.io/en/master](https://stable-baselines3.readthedocs.io/en/master)
8.1.1 Environments

Each gridworld is composed of an \( n \times m \) grid of states, \( S = \{0, \ldots, n-1\} \times \{0, \ldots, m-1\} \), along with a designated start state \( s_{\text{start}} \), designated goal state \( s_{\text{goal}} \), and a set \( B \subset S \) of blocked states which the agent is not permitted to enter. Episodes are of fixed length \( K \), and the agent begins each episode in state \( s_{\text{start}} \).

In a given state \( s = (i, j) \), the agent chooses an action \( a \in \{\text{stay}, \text{up}, \text{down}, \text{left}, \text{right}\} \). The agent then attempts to move in the direction corresponding to the action selected: if the selected action would move the agent off the grid or into a blocked state, the agent remains in \( s \); otherwise, the agent moves into (or remains in) the state corresponding to the action selected. For example, if \( a = \text{up} \) is chosen, the agent attempts to move to state \( s' = (i, j-1) \). If \( s' \) is off the grid (i.e. \( j-1 < 0 \)) or \( s' \in B \), the agent remains in \( s \). Otherwise, the agent transitions to \( s' \). Finally, let \( A(s) \) denote the set of all actions at \( s \) that do not lead off the grid or into a blocked state; the cost function is then given by:

\[
e(s, a) = \begin{cases} 
  c_{\text{goal}} & \text{if } s = s_{\text{goal}} \text{ and } a \in A(s), \\
  c_{\text{allowed}} & \text{if } s \neq s_{\text{goal}} \text{ and } a \in A(s), \\
  c_{\text{blocked}} & \text{if } a \notin A(s),
\end{cases}
\]

where \( 0 < c_{\text{goal}} < c_{\text{allowed}} < c_{\text{blocked}} \). A policy minimizing \( J(\theta) \) will move as quickly as possible to \( s_{\text{goal}} \) while always choosing actions within \( A(s) \). Because of this, when a problem is small enough that the agent can reach the goal state quickly and remain in it for most of the episode, the optimal average cost should be close to 1. A policy minimizing \( \rho(\theta) \), on the other hand, will seek to balance minimizing \( J(\theta) \) with maximizing \( H(d_{\theta}) \), while avoiding actions \( a \notin A(s) \).

For the first set of experiments, we considered the Gridworld1, 2, and 3 environments shown in Figure 3. The LargeGridWorld environment used in the second experiment is also depicted in Figure 3.

8.1.2 Implementation

For the first set of experiments, we implemented a tabular version of Algorithm 2. In order to have a baseline to compare against, we also implemented classic average-cost actor-critic, vanilla AC. Pseudocode for vanilla AC is included in the appendix. For both algorithms, we used tabular softmax policies:

\[
\pi_\theta(a_i | s) = \frac{\exp(\theta^T \psi(s, a_i))}{\sum_j \exp(\theta^T \psi(s, a_j))},
\]

where \( \theta \in \mathbb{R}^{\lvert S \rvert \times \lvert A \rvert} \) and \( \psi: S \times A \to \mathbb{R}^{\lvert S \rvert \times \lvert A \rvert} \) maps each state-action pair to a unique standard basis vector \( e_k \in \mathbb{R}^{\lvert S \rvert \times \lvert A \rvert} \), where \( e_k \) has a 1 in its \( k \)th entry and 0 everywhere else. We similarly used tabular representation.
Figure 3: GridWorld environments. For Gridworlds 1, 2, and 3, the start state is S and goal state is G. Shaded regions represent blocked states B. For the LargeGridWorld environment, the blue square is the start state and the green square is the goal state.
for the value functions:

\[ v_\omega(s) = \omega^T \phi(s), \]

where \( \omega \in \mathbb{R}^{|S|} \) and \( \phi : S \to \mathbb{R}^{|S|} \) maps each state \( s_i \) to a unique standard basis vector \( e_i \).

For the second set of experiments, we implemented IDAC with a categorical policy using two-layer, fully connected neural networks for both the policy and value function approximators, and we compared against the Stable Baselines 3 implementations of A2C, DQN, and PPO with two-layer, fully connected neural networks for all policies and value function approximators.

8.1.3 Tabular Experiment Results

Figures 4, 5, and 6 compare IDAC and vanilla AC on the GridWorld environments with \( c_{\text{goal}} = 1 \), \( c_{\text{allowed}} = 10 \), and \( c_{\text{blocked}} = 100 \). To generate these figures, 15 instances of each algorithm were run on the environment, the average cost and entropy were computed for each episode, and the sample means and 95% confidence intervals for the cost, entropy, and corresponding OIR over the 15 runs were used to generate the learning curves. As the figures show, the OIR algorithm outperforms the vanilla algorithm in every case. In particular, IDAC consistently explores the state space, leading to eventual discovery of the goal state, while vanilla AC quickly becomes deterministic and converges to a suboptimal policy. This supports our interpretation that, in the sparse-reward setting, algorithms minimizing the OIR fall back on maximizing a useful secondary objective, \( H(d_\theta) \); this can provide an advantage over vanilla methods focused solely on minimizing \( J(\theta) \).

On all three GridWorld environments, for both the IDAC and vanilla AC algorithms we used actor learning rate \( \alpha = 1.8 \), critic learning rate \( \beta = 2.0 \), and geometric mixing rate \( \tau = 0.1 \). For IDAC, we set \( \kappa = 1.0 \). We chose these parameters through trial and error. For GridWorld1, we used episode length 200 over 2500 episodes. For GridWorld2, we used episode length 200 over 3000 episodes. Finally, for GridWorld3, we used episode length 300 over 3000 episodes. To facilitate learning, we found it helpful to increase the episode length and number of episodes as the complexity of the problem increased. Figures 4, 5, and 6 present the average cost, entropy, and OIR (with \( \kappa = 1.0 \)) for both algorithms as training proceeds. Note that we did not provide optimal benchmarks for these problems using the concave program solver. Since the solver finds the optimal state-action occupancy measure based on the assumption that it is independent of the initial start state, the designated start states inherent in the GridWorld environments causes the solver results to be inaccurate.
As can be seen from the average cost in all three figures, both algorithms quickly learn to avoid actions moving off the grid or into blocked states, decreasing to an average cost of around 10. On all three environments, vanilla AC gets stuck near 10 for the remainder of training. This corresponds to taking allowed actions, but not attaining the goal state. The IDAC algorithm, on the other hand, clearly spends an increasing amount of time in the goal state, since its cost decreases well below 10. Next, the evolution of the state occupancy measure entropy achieved by the two algorithms during training is similar to that seen in the SimpleEnv experiment. Vanilla actor-critic converges fairly quickly to a policy visiting only a small subset of the available states, reflecting overconfidence in its past experience. This is why vanilla AC struggles on these environments, since its policy becomes deterministic before the state space has been sufficiently explored. IDAC, in contrast, maintains policies with relatively high state occupancy measure entropy early on, only decreasing as the algorithm seeks to strike the right balance between cost and entropy. Finally, in all cases IDAC makes clear progress minimizing the OIR, while vanilla AC consistently increases it.

8.1.4 Neural Network Experiment Results

Figures 1 and 2 illustrate the performance of neural IDAC and A2C, DQN, and PPO on LargeGridWorld with $c_{\text{goal}} = 0.1$, $c_{\text{allowed}} = 5$, and $c_{\text{blocked}} = 10$. To generate the data for these figures, we first trained 48 instances of neural IDAC with different random seeds. We next trained 15 instances of each of the A2C, DQN, and PPO algorithms on the environment. For each algorithm, the average cost was computed for each episode, and the sample means and 95% confidence intervals were used to create the learning curves. As Figure 2 illustrates, 40 of the IDAC trials succeeded in finding the goal state, while 8 failed. To create Figure 1, we randomly selected 15 of the 40 successful runs of IDAC to compare with A2C, DQN, and PPO. As the figure illustrates, IDAC outperformed all three. Furthermore, none of A2C, DQN, and PPO found the goal state after $1 \times 10^6$ timesteps.

Hyperparameters $\alpha = 0.0001$, $\beta = 0.0002$, $\tau = 0.1$, and $\kappa = 0.1$ for neural IDAC were selected through trial and error. After finding that increasing the width of the layers improved performance, we used 512 hidden units for each layer in both the policy and value functions. After experimenting with a range of different parameters and detecting no noticeable difference in performance, Stable Baselines’ default parameters for A2C, DQN, and PPO were used. This included learning rates 0.0007 for A2C, 0.0003 for PPO, and 0.0001 for DQN, as well as 64-width layers for all networks.

As in the tabular experiments, all algorithms quickly learn to avoid blocked actions. In the case of A2C and PPO, this leads to an average cost of exactly 5, while for DQN the cost remains slightly above 5 due to exploration noise lower bounded by 0.05. Though the optimal cost is 0.1, once they have converged to these values, they remain there for the remainder of training. Once again, the combination of sparse reward signals and overconfidence in past experience likely caused this premature convergence. Meanwhile, since neural IDAC is minimizing $\rho(\theta)$ instead of $J(\theta)$, it swiftly locates the goal state and finds an optimal policy with average cost 0.1. This illustrates that, in sparse-reward environments, OIR-based policy gradient methods can lead to improved performance over vanilla techniques.

8.2 Illustration of Theoretical Results

To illustrate the theory developed above, we compared the results of training the IDAC algorithm analyzed in Section 7 with the optimal solutions predicted by solving the concave program described in Section 5. Our
experiments show that, on a simple problem in the tabular setting, our algorithm converges to the optimal OIR obtained by solving the concave program. We also compare with vanilla AC to demonstrate that taking entropy into account causes IDAC to solve a different problem than that solved by vanilla AC. Finally, we examine how the optimal solution to the concave program changes as a function of $\kappa$ in order to illustrate the role of $\kappa$ as a regularizer in the OIR (6).

8.2.1 Environment

We considered a five-state, five-action environment with deterministic dynamics: in a given state $s \in S = \{0, 1, 2, 3, 4\}$, choosing action $a \in A = \{0, 1, 2, 3, 4\}$ causes the agent to transition directly to state $s = a$. This environment, called SimpleEnv, is depicted in Figure 7. The cost function $c : S \times A \rightarrow \{1, 2\}$ assigns $c(0, a) = 1$, for all $a \in A$, and $c(s, a) = 2$, for all $s \in \{1, 2, 3, 4\}$ and $a \in A$. Each episode starts in a random initial state and lasts for exactly $K$ timesteps before terminating. When seeking to minimize $J(\theta)$, the optimal policy always chooses action $a = 0$, which amounts to remaining in state $s = 0$. The optimal policy minimizing $\rho(\theta)$, on the other hand, will seek to balance minimizing $J(\theta)$ with maintaining sufficient state-space coverage $H(d_\theta)$, and it will also depend on $\kappa$. Note that SimpleEnv is equivalent to a multi-armed bandit problem with action set $A = S$.

8.2.2 Implementation

For these experiments we used the same tabular implementation of IDAC and vanilla AC described in the preceding subsection. We implemented the SimpleEnv environment as an MDP by specifying the corresponding cost vector $c \in \mathbb{R}^{|S| \times |A|}$ and transition probability function $p : S \times A \rightarrow \mathcal{D}(S)$. Based on the concave reformulation detailed in Section 5.1, we also implemented a solver for the minimizing the OIR over a given MDP using the CVXPY convex optimization package. When provided with a cost vector $c$, transition probability function $p$, and $\kappa \geq 0$, the solver returns the optimal OIR and the corresponding cost, entropy, state-action occupancy measure, state occupancy measure, and policy, as well as the optimal average cost for the underlying average-cost MDP. We used this solver to obtain the optimal OIR, the corresponding average cost and entropy, and the underlying optimal average cost for the SimpleEnv environment for various $\kappa$.

8.2.3 Experiment Results

We trained IDAC and vanilla AC on SimpleEnv for three different values of $\kappa$, then compared the results with the optimal values predicted by the solver. For both algorithms we used policy learning rate $\alpha = 0.5$, value function learning rate $\beta = 1.0$, and geometric mixing rate $\tau = 0.1$. Episode length was 200 and training took place over 1000 episodes. Since the primary goal was to illustrate the underlying theory on a simple problem rather than examine performance on an existing benchmark, we chose hyperparameters by trial and error instead of using a more rigorous grid search. We also ran the solver for a range of different values of $\kappa$ to illustrate how the OIR and associated average cost and entropy vary as a function of $\kappa$.

Figures 9, 10, and 11 compare the performance of IDAC and vanilla AC on the SimpleEnv environment, and they also compare each algorithm’s performance with the optimal benchmark provided by the solver. As the figures show, both algorithms approach the optimal values predicted by the solver, illustrating the correctness of the theory developed in preceding sections. Furthermore, IDAC and vanilla AC converge to distinct values, highlighting the fact that taking entropy into account fundamentally alters the underlying problem being solved.

*www.cvxpy.org*
To generate these figures, 15 instances of each algorithm were run on the environment, the empirical average cost and entropy were computed for each episode, and the sample means and 95% confidence intervals for the cost, entropy, and corresponding OIR over the 15 runs were used to generate the learning curves. In Figures 9 and 10, both algorithms clearly converge to the global optimum predicted by the solver of their respective problems: (6) in the case of IDAC, and (3) in the case of vanilla AC. In Figure 11, both algorithms converge to near the global optimum, but fall just short of achieving optimality; we hypothesize that the relative difficulty of the problem means that further training is needed. As expected, in all cases vanilla AC becomes deterministic quickly, while IDAC learns a policy balancing somewhat higher cost with a higher-entropy occupancy measure. Also note how the OIR of the policy yielded by vanilla AC becomes larger, while that yielded by IDAC decreases to the optimum. This is expected, since vanilla AC ignores $H(d_\theta)$, focusing solely on minimizing $J(\theta)$.

Figure 8 shows the optimal OIR and the corresponding cost and entropy as a function of $\kappa$. These values were generated using the concave program solver described above. Recall from Remark 1 that $\kappa$ acts as a regularizer for the OIR, $\rho(\theta)$, by adjusting the importance of the entropy $H(d_\theta)$: when $\kappa$ is small, minor changes in the value of $H(d_\theta)$ will have a large effect on the value of $\rho(\theta)$, so larger values of $H(d_\theta)$ are encouraged when minimizing $\rho(\theta)$; when $\kappa$ is large, on the other hand, significant perturbations of the value of $H(d_\theta)$ will have little effect on the value of $\rho(\theta)$, so it is more important to have smaller values of $J(\theta)$. Figure 8 illustrates this phenomenon. For small values of $\kappa$, greater emphasis is placed on entropy, so the entropy $H(d_\theta^*)$ corresponding to the optimal $\theta^*$ is large, while $J(\theta^*)$ is large as well. As $\kappa$ grows, greater emphasis is placed on minimizing cost, so the corresponding cost $J(\theta^*)$ decreases, while the entropy $H(d_\theta^*)$ diminishes as well.

Figure 8: Optimal OIR and corresponding cost and entropy as functions of $\kappa$.

Figure 9: Learning curves for vanilla AC and IDAC with $\kappa = 0.5$. 
9 Conclusion

In this paper we have addressed the exploration/exploitation trade-off in reinforcement learning by developing policy gradient methods that optimize exploitation per unit exploration via a new RL objective: the occupancy information ratio, or OIR. En route, we have elaborated a rich theory underlying these methods, including: a concave programming reformulation of the OIR optimization problem with links to the powerful linear programming theory for MDPs; policy gradient theorems for the OIR setting, including entropy gradient theorems of independent interest; both asymptotic and non-asymptotic convergence theory with global optimality guarantees. We have furthermore presented empirical results that both validate the theory and indicate promising empirical performance when compared with state-of-the-art methods on sparse-reward problems. Interesting directions for future work include extension of our theoretical results to more general classes of ratio optimization problems, exploration of max-entropy policy gradient methods leveraging our entropy gradient theorems, development of variants of the Information-Directed Actor-Critic algorithm for continuous spaces using suitable density estimation techniques, and thorough empirical evaluation of deep RL variants of IDAC on a broad class of benchmark problems.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24:2312–2320, 2011.

[2] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.

[3] Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39.1–39.26. PMLR, 2012.

[4] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.
[5] Karl Johan Åström and Peter Eykhoff. System identification – a survey. *Automatica*, 7(2):123–162, 1971.

[6] Mordecai Avriel, Walter E. Diewert, Siegfried Schaible, and Israel Zang. *Generalized Concavity*. SIAM, 2010.

[7] Amrit Singh Bedi, Anjaly Parayil, Junyu Zhang, Mengdi Wang, and Alec Koppel. On the sample complexity and metastability of heavy-tailed policy search in continuous control. *arXiv preprint arXiv:2106.08414*, 2021.

[8] Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.

[9] Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

[10] Shalabh Bhatnagar, Richard Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[11] Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[12] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. $\mathcal{X}$-armed bandits. *Journal of Machine Learning Research*, 12(5):1655–1695, 2011.

[14] Hiroki Furuta, Tatsuya Matsushima, Tadashi Kozuno, Yutaka Matsuo, Sergey Levine, Ofir Nachum, and Shixiang Shane Gu. Policy information capacity: Information-theoretic measure for task complexity in deep reinforcement learning. *arXiv preprint arXiv:2103.12726*, 2021.

[15] John C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

[16] Robert M. Gray. *Entropy and Information Theory*. Springer Science & Business Media, 2011.

[17] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[18] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

[19] Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I. Jordan. Is Q-learning provably efficient? *Advances in Neural Information Processing Systems*, 2018.

[20] Sham M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14, 2001.

[21] Vijaymohan Konda. *Actor-Critic Algorithms*. PhD thesis, MIT, 2002.

[22] Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Science & Business Media, 1978.

[23] Harold J. Kushner and George G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic Modelling and Applied Probability. Springer-Verlag New York, 2003.

[24] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[25] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.
[26] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[27] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[28] Xiujuan Lu, Benjamin Van Roy, Vikranth Dwaracherla, Morteza Ibrahimi, Ian Osband, and Zheng Wen. Reinforcement learning, bit by bit. *arXiv preprint arXiv:2103.04047*, 2021.

[29] Jincheng Mei, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[31] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2003.

[32] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[33] Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Advances in Neural Information Processing Systems*, 27:1583–1591, 2014.

[34] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of Thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[35] Siegfried Schaible. Fractional programming. I, duality. *Management Science*, 22(8):858–867, 1976.

[36] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[38] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[39] Wesley A. Suttle, Kaiqing Zhang, Zhuoran Yang, David Kraemer, and Ji Liu. Reinforcement learning for cost-aware Markov decision processes. In *International Conference on Machine Learning*, pages 9989–9999. PMLR, 2021.

[40] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[41] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 99:1057–1063, 1999.

[42] Nassim Nicholas Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007.

[43] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[44] John N. Tsitsiklis and Benjamin Van Roy. Average cost temporal-difference learning. *Automatica*, 35(11):1799–1808, 1999.

[45] Peter Whittle. Multi-armed bandits and the Gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):143–149, 1980.
A Entropy Maximization Algorithms

A.1 Maximizing $H(d_{\theta})$

Algorithms for maximizing $H(d_{\theta})$ based on Theorem 2 are given below. These schemes can be easily combined with standard policy gradient algorithms to provide a state exploration bonus.

Algorithm 3: MaxStateEntropyREINFORCE

Data: Rollout length $K$, stepsizes $\eta < \tau$, parametrized policy class $\pi_{\theta}$

1 Initialization: randomly sample $\theta_0$; select $\bar{\mu}_0 > 0$; $t \leftarrow 0$

2 while $0 < 1$ do

3 Generate trajectory $\{(s_i, a_i)\}_{i=1,...,K}$ using policy $\pi_{\theta_t}$

4 Compute estimate $\hat{d}_{\theta_t}(s)$, for each $s \in S$ encountered

5 Compute estimate $\hat{\mu}_t(\theta_t) = \frac{1}{K} \sum_{i=1}^{K} \left(-\log \hat{d}_{\theta_t}(s_i)\right)$

6 Update $\bar{\mu}_t = (1 - \tau) \bar{\mu}_{t-1} + \tau \hat{\mu}_t(\theta_t)$

7 Compute $\nabla H(d_{\theta_t}) = \frac{1}{K} \sum_{i=1}^{K} \left[-\log \hat{d}_{\theta_t}(s_i) - \bar{\mu}_t\right] \nabla \log \pi_{\theta_t}(a_i|s_i)$

8 $\theta_{t+1} = \theta_t + \eta \nabla H(d_{\theta_t})$

9 $t \leftarrow t + 1$

end

A.2 Maximizing $H(\lambda_{\theta})$

Algorithms for maximizing $H(\lambda_{\theta})$ based on Corollary 1 are given next. Again, these schemes can be easily combined with standard policy gradient algorithms to provide a state-action exploration bonus. Note that, though we are estimating the entropy of the state-action occupancy measure, we only explicitly learn an approximation for the lower-dimensional state occupancy measure $d_{\theta}$. 

[46] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

[47] Junyu Zhang, Amrit Singh Bedi, Mengdi Wang, and Alec Koppel. Beyond cumulative returns via reinforcement learning over state-action occupancy measures. In *2021 American Control Conference*, pages 894–901, 2021.

[48] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvári, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33:4572–4583, 2020.

[49] Kaiching Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.
Algorithm 4: MaxStateEntropyActorCritic

Data: Rollout length $K$, stepsizes $\tau$ and $\alpha < \beta$, parametrized policy class $\pi_\theta$, parametrized critic class $v_\omega$

Initialization: randomly sample $\theta_0, \omega_0$; select $\tilde{\mu}_0 > 0$; $t \leftarrow 0$

while $0 < 1$

Generate trajectory $\{(s_i, a_i)\}_{i=1}^K$ using policy $\pi_{\theta_t}$

Compute estimate $\hat{d}_{\theta_t}(s)$, for each $s \in S$ encountered

$\hat{\mu}^t(\theta_t) = \frac{1}{K} \sum_{i=1}^K \left( -\log \hat{d}_{\theta_t}(s_i) \right)$

$\tilde{\mu}_t = (1 - \tau) \tilde{\mu}_{t-1} + \tau \hat{\mu}^t(\theta_t)$

for $i = 1, \ldots, K$

Compute TD error

$\delta_i = -\log \hat{d}_{\theta_t}(s_i) - \tilde{\mu}_t + v_\omega(s_{i+1}) - v_\omega(s_i)$

where we stipulate $v_\omega(s_{K+1}) = 0$

end

$\tilde{\nabla} v_\omega = \frac{1}{K} \sum_{i=1}^K \delta_i \tilde{\nabla} v_\omega(s_i)$

$\omega_{t+1} = \omega_t + \beta \tilde{\nabla} v_\omega$

$\nabla H(d_{\theta_t}) = \frac{1}{K} \sum_{i=1}^K \delta_i \nabla \log \pi_{\theta_t}(a_i|s_i)$

$\theta_{t+1} = \theta_t + \alpha \nabla H(d_{\theta_t})$

$t \leftarrow t + 1$

end

Algorithm 5: MaxStateActionEntropyREINFORCE

Data: Rollout length $K$, stepsizes $\eta < \tau$, parametrized policy class $\pi_\theta$

Initialization: randomly sample $\theta_0$; select $\tilde{\mu}_0 > 0$; $t \leftarrow 0$

while $0 < 1$

Generate trajectory $\{(s_i, a_i)\}_{i=1}^K$ using policy $\pi_{\theta_t}$

Compute estimate $\hat{d}_{\theta_t}(s)$, for each $s \in S$ encountered

Compute estimate $\mu^t(\theta_t) = \frac{1}{K} \sum_{i=1}^K \left( -\log \hat{d}_{\theta_t}(s_i) - \log \pi_{\theta_t}(a_i|s_i) \right)$

Update $\tilde{\mu}_t = (1 - \tau) \tilde{\mu}_{t-1} + \tau \mu^t(\theta_t)$

Compute $\nabla H(\lambda_{\theta_t}) = \frac{1}{K} \sum_{i=1}^K \left[ -\log \hat{d}_{\theta_t}(s_i) - \log \pi_{\theta_t}(a_i|s_i) - \tilde{\mu}_t \right] \nabla \log \pi_{\theta_t}(a_i|s_i)$

$\theta_{t+1} = \theta_t + \eta \nabla H(\lambda_{\theta_t})$

$t \leftarrow t + 1$

end
Algorithm 6: MaxStateActionEntropyActorCritic

Data: Rollout length $K$, stepsizes $\tau$ and $\alpha < \beta$, parametrized policy class $\pi_\theta$, parametrized critic class $v_\omega$

1 Initialization: randomly sample $\theta_0, \omega_0$; select $\bar{\mu}_0 > 0$; $t \leftarrow 0$
2 while $0 < 1$ do
3     Generate trajectory $\{(s_i, a_i)\}_{i=1,\ldots,K}$ using policy $\pi_\theta$
4     Compute estimate $\hat{d}_\theta(s)$, for each $s \in S$ encountered
5     $\mu^x(\theta_t) = \frac{1}{K} \sum_{i=1}^K \left( -\log \hat{d}_\theta(s_i) - \log \pi_\theta(a_i | s_i) \right)$
6     $\bar{\mu}_t = (1 - \tau)\bar{\mu}_{t-1} + \tau \mu^x(\theta_t)$
7     for $i = 1, \ldots, K$ do
8         Compute TD error
9         $\delta_i = -\log \hat{d}_\theta(s_i) - \log \pi_\theta(a_i | s_i) - \bar{\mu}_t + v_\omega(s_{t+1}) - v_\omega(s_t)$
10        where we stipulate $v_\omega(s_{K+1}) = 0$
11     end
12     $\nabla v_\omega = \frac{1}{K} \sum_{i=1}^K \delta_i \nabla v_\omega(s_i)$
13     $\omega_{t+1} = \omega_t + \beta \nabla v_\omega$
14     $\nabla H(\lambda_\theta) = \frac{1}{K} \sum_{i=1}^K \delta_i \nabla \log \pi_\theta(a_i | s_i)$
15     $\theta_{t+1} = \theta_t + \alpha \nabla H(\lambda_\theta)$
16     $t \leftarrow t + 1$
17 end

Algorithm 7: VanillaActorCritic

Data: Rollout length $K$, stepsizes $\tau$ and $\alpha < \beta$, parametrized policy class $\pi_\theta$, parametrized critic class $v_\omega$

1 Initialization: randomly sample $\theta_0, \omega_0$; select $\bar{\mu}_0 > 0$; $t \leftarrow 0$
2 while $0 < 1$ do
3     Generate trajectory $\{(s_i, a_i)\}_{i=1,\ldots,K}$ using policy $\pi_\theta$
4     $\hat{J}(\theta_t) = \frac{1}{K} \sum_{i=1}^K c(s_i, a_i)$
5     $\bar{\mu}_t = (1 - \tau)\bar{\mu}_{t-1} + \tau \hat{J}(\theta_t)$
6     for $i = 1, \ldots, K$ do
7         Compute TD error:
8         $\delta_i = c(s_i, a_i) - \bar{\mu}_t + v_\omega(s_{t+1}) - v_\omega(s_t)$
9         where we stipulate $v_\omega(s_{K+1}) = 0$
10     end
11     $\nabla v_\omega = \frac{1}{K} \sum_{i=1}^K \delta_i \nabla v_\omega(s_i)$
12     $\omega_{t+1} = \omega_t + \beta \nabla v_\omega$
13     $\nabla J(\theta_t) = \frac{1}{K} \sum_{i=1}^K \delta_i \nabla \log \pi_\theta(a_i | s_i)$
14     $\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$
15     $t \leftarrow t + 1$
16 end

B Pseudocode for VanillaActorCritic

For completeness, we provide pseudocode for the version of average-cost actor-critic that we compared against in our experimental results: