Active Model Estimation in Markov Decision Processes

Jean Tarbouriech\textsuperscript{1,2} Shubhanshu Shekhar\textsuperscript{3} Matteo Pirotta\textsuperscript{1} Mohammad Ghavamzadeh\textsuperscript{1} Alessandro Lazaric\textsuperscript{1}

Abstract

We study the problem of efficient exploration in order to learn an accurate model of an environment, modeled as a Markov decision process (MDP). Efficient exploration in this problem requires the agent to identify the regions in which estimating the model is more difficult and then exploit this knowledge to collect more samples there. In this paper, we formalize this problem, introduce the first algorithm to learn an $\epsilon$-accurate estimate of the dynamics, and provide its sample complexity analysis. While this algorithm enjoys strong guarantees in the large-sample regime, it tends to have a poor performance in early stages of exploration. To address this issue, we propose an algorithm that is based on maximum weighted entropy, a heuristic that stems from common sense and our theoretical analysis. The main idea here is cover the entire state-action space with the weight proportional to the noise in the transitions. Using a number of simple domains with heterogeneous noise in their transitions, we show that our heuristic-based algorithm outperforms both our original algorithm and the maximum entropy algorithm in the small sample regime, while achieving similar asymptotic performance as that of the original algorithm.

1 INTRODUCTION

In most decision problems, the agent is provided with a goal that it tries to achieve by maximizing a reward signal. In such problems, the agent explores the environment in order to identify the high reward situations and reach the goal faster and more efficiently. Although solving a problem (achieving a goal) is usually the ultimate objective, it is sometimes equally important for an agent to understand its environment without pursuing any goals. In such scenarios, no reward function is defined and the agent explores the state-action space in order to discover what is possible and how the environment works. We refer to this scenario as reward-free or unsupervised exploration. Several objectives have been studied in reward-free exploration, including discovering incrementally reachable states \cite{14}, uniformly covering the state space \cite{10}, estimating state-dependent random variables \cite{21}, a broad class of objectives that are defined as functions of the state visitation frequency induced by the agent’s behavior \cite{6,7}, and learning a model of the environment that is suitable for computing near-optimal policies for a given collection of reward functions \cite{12}. In all these cases, a reward-free exploration algorithm is evaluated by the amount of exploration it uses to learn its objective.

None of the works above, except \cite{12}, focuses on learning the dynamics of an environment modeled as a Markov decision process (MDP). Even in \cite{12}, the setting is the simpler finite-horizon MDP, where many states are often irrelevant as they have no impact in defining the optimal policy for any reward functions. Although the problem of learning the dynamics has not been rigorously studied in the reward-free exploration setting, \cite{11} proposed several heuristics for this problem. Studying active exploration for model estimation is also important in the theoretical understanding of the simulation-to-real problem, where the goal is to start with an inaccurate model (simulator) of the environment and learn a better one with minimum interaction with the world.

In this paper, we formalize the problem of reward-free exploration where the objective is to estimate both a uniformly accurate (minimizing the maximum error) and an average accurate (minimizing the average error) model of the environment, i.e., the transition probability function of the MDP. We identify the form of the model estimation
error for a given policy and show that it depends on how
often the policy visits noisy state-action pairs, i.e., those
whose transition probability has high variance. Since op-
timizing the model estimation error over the policies is
difficult, we upper-bound it using Bernstein’s inequality
and obtain an objective function that relates the structure
of the MDP with the accuracy of a model estimated by a
certain state-action visitation (stationary distribution of a
policy). We then build on [21] and propose an algorithm
that optimizes this objective function over stationary dis-
tributions and prove sample complexity bounds for the
accuracy (both on average and in worst case) of the es-
estimated model. In particular, our analysis highlights the
intrinsic difficulty of the model estimation problem.

Our “exact” algorithm may be inefficient due to the spe-
cific form of the objective function we optimize and it
tends to perform well only in large sample regimes
(asymptotically). Furthermore, our theoretical guarantees
only hold under restrictive assumptions on the MDP (i.e.,
ergodicity). To alleviate these limitations, we replace the
objective function used by our algorithm with maximizing
weighted entropy, where the goal is to visit state-action
pairs weighted by the noise in their transitions. Although
this is a heuristic, it stems from our derived objective
function and is more tailored to the model estimation problem
than the popular maximum entropy objective [10, 6] that
uniformly covers the state space. We derive an algorithm
based on the maximum weighted entropy heuristic, by
modifying and extending the algorithm in [6], and we
prove regret guarantees w.r.t. the stationary distribution
maximizing the weighted entropy.

The main contributions of this paper are: 1) We introduce
an objective function and use it to derive and analyze
an algorithm for accurate MDP model estimation in the
reward-free exploration setting. The algorithm extends
the one in [21] by removing the requirement of knowing
the model. 2) When we switch from our original objective
function to the maximum weighted entropy heuristic, we
show how the algorithm in [6] can be modified to handle
an unknown objective (since the weights, i.e., noise in
transitions are not known in advance). 3) We show that
the algorithm based on the maximum weighted entropy
heuristic outperforms both our theoretically driven al-
gorithm and the maximum entropy algorithm in the small
sample regime, while achieving similar asymptotic per-
formance as that of our exact algorithm. Finally, 4) we
consider an undiscounted infinite-horizon setting, which
is more general than the finite horizon MDP setting con-
sidered in the recent work [12], where a large portion of
the states can be considered as irrelevant.

---

2 PROBLEM FORMULATION

We consider a reward-free finite Markov Decision Process
(MDP) $M := (S, A, p)$, where $S$ is the set of states, $A$
is the set of actions. Each state-action pair $(s, a) \in S \times
A$ is characterized by an unknown transition probability
function $p(\cdot | s, a)$ over next states. We denote by $\pi$
a stochastic non-stationary policy that maps a history
$(s_0, a_0, s_1, \ldots, s_t)$ of states and actions observed up to
time $t$ to a distribution over actions $A$. We also consider
stochastic stationary policies $\pi : S \to \Delta(A)$, which map
the current state $s_t$ to a distribution over actions.

At a high-level, the objective of the agent is to estimate
as accurately as possible the unknown transition dynam-
ics. We refer to this objective as the model estimation
problem, or ModEst for short. More formally, the agent
interacts with the environment by following a possibly
non-stationary policy $\pi$ and, after $n$ time steps, it returns
an estimate $\hat{p}_{\pi,n}$ of the transition dynamics. We evaluate
the accuracy of $\hat{p}_{\pi,n}$ in terms of its $\ell_1$-norm distance (i.e.,
total variation) w.r.t. the model $p$ either on average or in
the worst-case over the state-action space:

$$E_{\pi,n} := \frac{1}{SA} \sum_{(s,a) \in S \times A} \| \hat{p}_{\pi,n}(\cdot | s, a) - p(\cdot | s, a) \|_1, \quad (1)$$

$$W_{\pi,n} := \max_{(s,a) \in S \times A} \| \hat{p}_{\pi,n}(\cdot | s, a) - p(\cdot | s, a) \|_1. \quad (2)$$

We say that an estimate $\hat{p}_{\pi,n}$ is $(\epsilon, \delta)$-accurate in the sense
of $E$ (resp. $W$) if $\mathbb{P}(E_{\pi,n} \leq \epsilon) \geq 1 - \delta$, i.e., it is $\epsilon$-accurate
with at least $1 - \delta$ probability (resp. $W_{\pi,n}$). We then measure
the sample complexity of a policy $\pi$ as follows.

**Definition 1.** Given an error $\epsilon > 0$ and a confidence
level $\delta \in (0, 1)$, the ModEst sample complexity of an
agent executing a policy $\pi$ is defined as

$$C^E_{\text{ModEst}}(\pi, \epsilon, \delta) := \min \left\{ n \geq 1 : \mathbb{P}(E_{\pi,n} \leq \epsilon) \geq 1 - \delta \right\},$$

$$C^W_{\text{ModEst}}(\pi, \epsilon, \delta) := \min \left\{ n \geq 1 : \mathbb{P}(W_{\pi,n} \leq \epsilon) \geq 1 - \delta \right\}.$$  

According to the definition above, the objective is to find
an algorithm that is able to return an $(\epsilon, \delta)$-accurate model
with as little number of samples as possible, i.e., minimize
the sample complexity. While the definition above follows
a standard PAC-formulation, it is possible to consider the
case when a fixed budget $n$ is provided in advance and
the agent should return a estimate which is as accurate
as possible with confidence $1 - \delta$. In the following, we
consider both settings.

**Discussion.** While both $E_{\pi,n}$ and $W_{\pi,n}$ effectively
formalize the objective of accurate estimation of the dynam-
ics, they have specific advantages and disadvantages.

If $W_{\pi,n}$ is below $\epsilon$, it is possible to compute near-optimal
policies for any reward function (see e.g., [13]).
Proposition 1. For any \( \gamma \in (0, 1) \) and reward function \( r(s, a) \in [0, 1] \), let \( \bar{\pi} \) be the infinite-horizon \( \gamma \)-discounted optimal policy computed using the dynamics \( \bar{\rho} \) returned after the MODEST phase. If \( \bar{\rho} \) is \((\epsilon, \delta)\)-accurate in the sense of \( \mathcal{W} \), then \( \bar{\pi} \) is guaranteed to be \( O((1 - \gamma)^{-2}) \)-optimal in the exact MDP with probability at least \( 1 - \delta \).

While this shows the advantage of targeting the worst-case frequency of transitions. After \( n \) steps, the optimality gap \( \pi, n \) makes it more challenging to optimize than \( E_{\pi, n} \). Furthermore, consider the case where one single state-action pair has a very noisy dynamics but is overall uninteresting (e.g., the so-called noisy TV case), then targeting the worst-case error may require the agent to spend most of its time in estimating the dynamics of that single state-action pair, instead of collecting information about the rest of the MDP. A similar issue may rise in large MDPs, where lowering the estimation error uniformly over the whole state-action space may be prohibitive and lead to extremely large sample complexity.

Alternative metrics other than \( \ell_1 \) could be used to measure the distance between \( \hat{\rho}_{\pi, n} \) and \( \rho \). For instance, given the categorical nature of the transition distributions, a natural choice is to use the KL-divergence. Nonetheless, it has been shown that the KL-divergence is not a well-suited metrics for active learning as it promotes strategies close to uniform sampling [19]. Furthermore, other metrics may not provide any performance guarantees such as Prop. 1.

One may wonder why we rely on Def. 1 to measure the accuracy, instead of the closely related objective defined in [21], where the expected estimation error is used in place of its high-probability version. A careful inspection of the analysis in [21] shows that they critically rely on the independence between the stochastic transitions in the MDP and the randomness in the observations collected in each state. This argument cannot be applied to our case, where the observations of interest are the next state \( s' \) generated during transitions \((s, a)\) to \( s'\). This makes transitions and observations intrinsically correlated and it prevents us from using their analysis.

While minimizing the sample complexity \( C_{\text{MODEST}} \) over policies \( \pi \) is a well-posed objective, the dependency on \( \pi \) is implicit and the space of non-stationary policies is combinatorial in states, actions and time, thus making it impossible to directly optimize for \( C_{\text{MODEST}} \). In the next section we refine the error definitions to derive an explicit convex surrogate for stationary policies.

2.1 A Surrogate Convex Objective Function

We focus on model estimates defined as the empirical frequency of transitions. After \( n \) steps, for any state-action pair \((s, a)\) and any next state \( s' \), we define

\[
\hat{\rho}_{\pi, n}(s'|s, a) := \frac{T_{\pi, n}(s, a, s')}{T_{\pi, n}(s, a)},
\]

where \( T_{\pi, n}(s, a) := \sum_{t=1}^{n} \mathbb{1}_{(s_t, a_t) = (s, a)} \) denotes the (random) number of times action \( a \) was taken in state \( s \) during the execution of policy \( \pi \) over \( n \) steps. Similarly, we define \( T_{\pi, n}(s, a, s') := \sum_{t=1}^{n} \mathbb{1}_{(s_t, a_t, s_{t+1}) = (s, a, s')} \) and we define \( x^+ := \max\{x, 1\} \).

We introduce the notion of “noise” associated to each state-action pair.

Definition 2. For each state-action pair \((s, a)\), we define the transitional noise at \((s, a)\) as

\[
V(s, a) := \sum_{s' \in S} \sqrt{\frac{\sigma^2_p(s'|s, a)}{\mathbb{S}}},
\]

where \( \sigma^2_p(s'|s, a) \) is the variance of the transition from \((s, a)\) to \( s'\), i.e., \( \sigma^2_p(s'|s, a) := p(s'|s, a)(1 - p(s'|s, a)) \).

Denote by \( \Gamma(s, a) := \|p(\cdot|s, a)\|_0 \) the support of \( p(\cdot|s, a) \), as well as \( \Gamma := \max_{s,a} \Gamma(s, a) \). From the Cauchy-Schwarz inequality, we have

\[
0 \leq V(s, a) \leq \sqrt{\Gamma(s, a) - 1}/\sqrt{\mathbb{S}} \leq 1,
\]

where the lower bound \( V(s, a) = 0 \) is achieved for all \((s, a)\) \( \in \mathcal{S} \times \mathcal{A} \) when \( M \) is a deterministic MDP.

We connect the error functions to a term depending on the transitional noise and the number of state-action visits.

Proposition 2. For any fixed \( n \) and any policy \( \pi \), with probability at least \( 1 - \delta/3 \),

\[
E_{\pi, n} \leq \frac{4}{\mathcal{S}\mathcal{A}} \log \left( \frac{6\mathcal{S}\mathcal{A}n}{\delta} \right) \sum_{s,a} \mathcal{F}(s, a; T_{\pi, n}),
\]

\[
W_{\pi, n} \leq 4 \log \left( \frac{6\mathcal{S}\mathcal{A}n}{\delta} \right) \max_{s,a} \mathcal{F}(s, a; T_{\pi, n}),
\]

where

\[
\mathcal{F}(s, a; T) := \frac{V(s, a)}{\sqrt{T(s, a) + 1}} + \frac{\mathcal{S}}{T(s, a) + 1}.
\]

This upper-bound provides a much more explicit dependency between the behavior of \( \pi \), the structure of the MDP, and the accuracy of the model estimation. In particular, it illustrates how rarely visited pairs with large variance lead to larger estimation errors.

Convex reparametrization. Prop. 2 suggests that \( E_{\pi, n} \) (resp. \( W_{\pi, n} \)) could be optimized by directly minimizing the average (resp. the maximum) of its upper-bound \( \mathcal{F}(s, a; T) \). While the function \( \mathcal{F} \) is convex in \( T \), the
constraint that \( T_{\pi,n} \) must be a valid state-action counter renders the optimization problem of minimizing \( F(T_{\pi,n}) \) non-convex. To circumvent this issue, we introduce the state-action stationary distribution \( \lambda_{\pi} \in \Delta(S \times A) \) of a stationary policy \( \pi \). Also known as an occupancy measure, \( \lambda_{\pi}(s, a) \) is the expected frequency with which action \( a \) is executed in state \( s \) while following policy \( \pi \). Let \( \Lambda^{(p)} \) be the set of state-action stationary distributions, i.e.,

\[
\Lambda^{(p)} := \{ \lambda \in \Delta(S \times A) : \forall s \in S, \sum_{b \in A} \lambda(s, b) = \sum_{(s', a) \in S \times A} p(s'|s, a) \lambda(s', a) \}. \]

We introduce a convenient function

\[
G_n(s, a; \lambda) := \frac{V(s, a)}{\sqrt{\lambda(s, a)} + \frac{1}{n} + \frac{1}{n}}, \]

and its associated functions

\[
\mathcal{L}^{E}_n(\lambda) := \frac{1}{SA} \sum_{s,a} G_n(s, a; \lambda), \]

\[
\mathcal{L}^{W}_n(\lambda) := \max_{s,a} G_n(s, a; \lambda). \]

Let the empirical state-action frequency at any time \( t \) be \( \tilde{\lambda}_{\pi,t}(s, a) = T_{\pi,t}(s, a)/t \), then it is easy to see that

\[
\mathcal{E}_{\pi,n} \leq \frac{4}{\sqrt{n}} \log \left( \frac{6SA_n}{\delta} \right) \mathcal{L}^{E}_n(\tilde{\lambda}_{\pi,n}), \]

\[
\mathcal{W}_{\pi,n} \leq \frac{4}{\sqrt{n}} \log \left( \frac{6SA_n}{\delta} \right) \mathcal{L}^{W}_n(\tilde{\lambda}_{\pi,n}). \]

This suggests that we could directly minimize \( \mathcal{E}_{\pi,n} \) (resp. \( \mathcal{W}_{\pi,n} \)) by optimizing the function \( \mathcal{L}^{E}_n \) (resp. \( \mathcal{L}^{W}_n \)) over \( \lambda \) in the set of stationary distributions \( \Lambda^{(p)} \) as follows

\[
\min_{\lambda \in \Lambda^{(p)}} \mathcal{L}^{E}_n(\lambda), \min_{\lambda \in \Lambda^{(p)}} \mathcal{L}^{W}_n(\lambda). \tag{1} \]

These functions are convex in both the objective function and the constraints and we denote by \( \lambda^{E}_{I,n} \) and \( \lambda^{W}_{I,n} \) their solutions. Roughly speaking, the optimal distributions should visit more often state-action pairs associated to a larger transitional noise.

Beside \( \mathcal{L}_n \), we also use closely related functions obtained as asymptotic relaxation for \( n \to \infty \) and define the optimization problems

\[
\min_{\lambda \in \Lambda^{(p)}} \mathcal{L}^{E}_\infty(\lambda) := \frac{1}{SA} \sum_{s,a} \frac{V(s, a)}{\sqrt{\lambda(s, a)}}, \tag{*} \]

\[
\min_{\lambda \in \Lambda^{(p)}} \mathcal{L}^{W}_\infty(\lambda) := \max_{s,a} \frac{V(s, a)}{\sqrt{\lambda(s, a)}}. \]

We refer to \( \lambda^{E}_{\infty} \) and \( \lambda^{W}_{\infty} \) the corresponding optimal state-action stationary distributions.

3 LEARNING TO OPTIMIZE ModEst

In this section, we build on [21, 4], introduce a learning algorithm for the ModEst problem and prove its sample complexity for both \( \mathcal{E} \) and \( \mathcal{W} \) error functions. As the structure of the algorithm is the same for both cases, we describe it in its general form.

3.1 Setting the Stage

Leveraging [1] to design an algorithm minimizing the estimation error requires addressing two issues: 1) the transitional noise \( V(s, a) \) in the objective function and the transition dynamics \( p \) appearing in the constraint are unknown, 2) despite \( \mathcal{L}^{E}_n \) and \( \mathcal{L}^{W}_n \) being convex, they are both poorly conditioned and difficult to optimize.

Parameters estimation. After \( t \) steps, the algorithm constructs estimate \( \hat{\lambda}_t \) as in Eq. 3 using the samples collected so far. Furthermore, it defines the empirical variance of the transition from \( (s, a) \) to \( s' \) as \( \hat{\sigma}_t^2(s'|s, a) := \hat{\lambda}_t(s'|s, a)(1 - \hat{\lambda}_t(s'|s, a)) \). Then we have the following high-probability guarantees on \( \hat{\lambda}_t \) and an upper-confidence bound on the transitional noise.

**Lemma 1.** Define \( B_t(s, a, s') := \{ \hat{p} \in \Delta(S) : |\hat{p}(s'|s, a) - \hat{\lambda}_t(s'|s, a)| \leq B_t(s, a, s') \} \) and

\[
B_t(s, a, s') := 2 \sqrt{T_t^+(s, a)} \frac{6\ell_t}{T_t^+(s, a)}, \]

where \( \ell_t := \log \left( \frac{6SAT_t^+(s, a)}{\delta} \right) \). Furthermore, let

\[
\hat{V}_t^+(s, a) := \frac{1}{\sqrt{S}} \sum_{s' \in S} \left( \sqrt{\hat{\sigma}_t^2(s'|s, a)} + \sqrt{2 \frac{\ell_t}{T_t^+(s, a)}} \right), \]

where \( \ell_t := \log \left( \frac{4SAT_t^+(s, a)}{\delta} \right) \). Then with probability at least \( 1 - \delta \), for any \( t > 0 \) and \( (s, a) \in S \times A \), we have

\[
V(s, a) \leq \hat{V}_t^+(s, a), \quad p(\cdot|s, a) \in B_t(s, a). \]

This result allows us to build upper bounds on the error functions and correctly manage the constraint set \( \Lambda^{(p)} \).

**Smooth optimization.** We first notice that both \( \mathcal{L}^{E}_n \) and \( \mathcal{L}^{W}_n \) are built on \( G_n \), which may grow unbounded when \( n \) grows to infinity and \( \lambda \) tends to zero in state-action pairs with non-zero transitional noise. Furthermore, the function is smooth with coefficient scaling as \( 1/n^{5/2} \), which tends to zero as \( n \to +\infty \). This means that optimizing \( \mathcal{L}_n \) functions may become increasingly more difficult for large \( n \). In order to avoid this issue we restrict the space of \( \lambda \). For any \( \eta > 0 \) we introduce the restricted simplex

\[
\Lambda^{(p)} := \{ \lambda \in \Lambda^{(p)} \mid \forall (s, a) \in S \times A, \lambda(s, a) \geq \eta \}. \]
Algorithm 1: FW- MODEST

1. **Input:** Constraint parameter \( \eta > 0 \), confidence \( \delta \in (0, 1) \), budget \( n \), objective function \( \mathcal{L}_n \) (either \( \mathcal{L}^W_n \) or \( \mathcal{L}^E_n \)), and empirical frequency \( \hat{\lambda}_n(s, a) := \frac{1}{n} \).
2. **Initialization:** Set \( t := 1 \), state-action counter \( T_n(s, a) := 0 \) and empirical frequency \( \hat{\lambda}_n(s, a) := \frac{1}{n} \).
3. **for** episode \( k = 1, 2, \ldots \) **do**
   1. Compute the upper-confidence bound \( \hat{V}_k^+ \) to the estimates of the transitional noise.
   2. Compute the objective function \( \mathcal{L}_n \) using \( \hat{V}_k^+ \).
   3. Compute the occupancy measure \( \phi_{k+1}^+ \) using (6).
   4. Derive the policy \( \pi_{k+1}^+ \) using (7).
   5. Execute \( \pi_{k+1}^+ \) for \( \tau_k \) steps.
   6. Update the time index \( t \) and empirical frequency \( \hat{\lambda}_{k+1} \).

On this set we have that \( \mathcal{G}_n \) is now bounded and smooth independently from \( n \). Since \( \mathcal{L}^E_n \) is just an average of functions \( \mathcal{G}_n \), it directly inherits these properties.

**Proposition 3.** For any \( \lambda \in \Lambda_n^{(p)} \), the function \( \mathcal{L}^E_n(\lambda) \) has a smoothing constant scaling as \( 1/\eta^3 \).

Unfortunately, the same guarantees do not hold for \( \mathcal{L}^W_n \), as it is a non-smooth function of \( \mathcal{G}_n \). We thus modify \( \mathcal{L}^W_n \) using a LogSumExp (LSE) transformation. We briefly recall its properties (see e.g., [3]).

**Proposition 4.** For any \( x \in (\mathbb{R}_+)^m \), let LSE \( x := \log \left( \sum_{i=1}^m \exp(x_i) \right) \). Then the following properties hold:

1) LSE is convex and 1-smooth w.r.t. the \( \ell_\infty \)-norm.

2) \( \max_{i \in [m]} x_i \leq \text{LSE}(x) \leq \max_{i \in [m]} x_i + \log(m) \).

While the first property ensures convexity and smoothness, the second statement shows that the bias introduced by the transformation is logarithmic in the number of elements. We define the LSE version of \( \mathcal{L}^W_n \) as

\[
\mathcal{L}^W_n(\lambda) := \log \left( \sum_{s,a} \exp(\mathcal{G}_n(s, a; \lambda)) \right),
\]

which satisfies the following property by the chain rule and Prop. 4.

**Proposition 5.** For any \( \lambda \in \Lambda_n^{(p)} \), the function \( \mathcal{L}^W_n(\lambda) \) has a smoothing constant scaling as \( 1/\eta^3 \).

Comparing Prop. 5 to Prop. 3, we notice that beside being biased, \( \mathcal{L}^W_n \) is also less smooth than \( \mathcal{L}^E_n \) and it is thus possibly harder to optimize.

### 3.2 The Algorithm

We build on the estimation algorithm of [21] and propose FW- MODEST (Alg. [1]), which proceeds through episodes \( k = 1, 2, \ldots \) and applies a Frank-Wolfe approach to minimize the model estimation error. The algorithm receives as input the state-action space, the parameter \( \eta \) used in defining the restricted space \( \Lambda \), the budget \( n \), and the objective function \( \mathcal{L}_n \) to optimize (i.e., either \( \mathcal{L}^E_n \) or the smoothed \( \mathcal{L}^W_n \)). We first review how \( \mathcal{L}_n \) could be optimized by the Frank-Wolfe (FW) algorithm in the exact case, when the transition dynamics \( p \) and transitional noise \( V(s, a) \) are known. Let \( \lambda_0 \) be the initial solution in \( \Lambda_n^{(p)} \), then FW proceeds through iterations by computing

\[
\phi_{k+1} = \arg \min_{\lambda \in \Lambda_n^{(p)}} \langle \nabla \mathcal{L}_n(\lambda_k), \lambda \rangle,
\]

which is then used to update the candidate solution as \( \lambda_{k+1} = \lambda_k + \beta_k (\phi_{k+1} - \lambda_k) \). Whenever the learning rate \( \beta_k \) is properly tuned, the overall process is guaranteed to converge to optimal value \( \mathcal{L}_n(\lambda_{k,n}) \) at the rate \( O(\kappa_n/k) \) where \( \kappa_n \) is the smoothness of the function (see Prop. 3). FW- MODEST integrates the Frank-Wolfe scheme into a learning loop, where the solution constructed by the agent at the beginning of episode \( k \) is the empirical frequency of visits \( \hat{\lambda}_k \) and the optimization in (5) is replaced by an estimated version where the upper bound \( \hat{V}_k^+ \) from Lem. 1 is used in place of the transitional noise \( V(s, a) \). Moreover, the confidence interval \( B_k \) over the transition dynamics is used to characterize the constraint \( \Lambda(p) \), thus leading to the optimization

\[
\hat{\phi}_{k+1}^+ = \arg \min_{\lambda \in B_k \Lambda_n^{(p)}} \langle \nabla \hat{\mathcal{L}}_{n,k}^+(\hat{\lambda}_k), \lambda \rangle,
\]

where \( \hat{\mathcal{L}}_{n,k}^+ \) is \( \mathcal{L}_n \) with \( V \) replaced by \( \hat{V}_k^+ \) in the formulation of \( \mathcal{G}_n \). The problem above can be seen as the problem of solving an MDP with an optimistic choice of the dynamics within the confidence set \( B_k \) where the reward is set to \( -\nabla \hat{\mathcal{L}}_{n,k}^+(\hat{\lambda}_k) \), i.e., state-action pairs with larger (negative) gradient have larger reward. This directly leads to solutions that try to increase the accuracy in state-action pairs where the current estimation of the model is likely to have a larger error. Notice that the output of the optimization is the distribution \( \hat{\phi}_{k+1}^+ \) which cannot be directly used to update the solution \( \hat{\lambda}_k \) as in FW, since the empirical frequency can only be updated by executing a policy collecting new samples. Thus, \( \hat{\phi}_{k+1}^+ \) is used to define a policy \( \pi_{k+1}^+ \) as

\[
\pi_{k+1}^+(a|s) := \frac{\hat{\phi}_{k+1}^+(s, a)}{\sum_{b \in A} \hat{\phi}_{k+1}^+(s, b)},
\]

which is then executed for \( \tau_k \) steps until the end of the current episode. The samples collected throughout the episode are then used to compute the new frequency \( \hat{\lambda}_{k+1} \).

\footnote{In the following we use \( \mathcal{L}_n \) for the generic function the algorithm is required to optimize.}

\footnote{The \( \arg \min \) extracts only the solution \( \lambda \).}
The extended LP problem. While the overall structure of FW-ModeEst is similar to the estimation algorithm in [21], the crucial difference is that the true dynamics \( p \) is unknown and \( \tilde{p} \) cannot be directly solved as an LP. In fact, while for any fixed \( \tilde{p} \), the set \( \Lambda^{(\tilde{p})} \) only poses linear constraints and optimizing over \( \lambda \) coincides with the standard dual LP to solve MDPs [17], in our case we have to list all possible values of \( \tilde{p} \) in \( \mathcal{B}_k \). We thus propose to rewrite \( \tilde{L}_k \) as an extended LP problem by considering the state-action next-state occupancy measure \( q(s, a, s') \), defined as \( q(s, a, s') := p(s'|s, a)\lambda(s, a) \).

We recall from Lem. 2 that at any episode \( k \), we have an estimate \( \hat{p}_k \) such that, w.h.p., for any \((s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \),

\[
|\hat{p}_k(s'|s, a) - p(s'|s, a)| \leq B_k(s, a, s'),
\]

For notational convenience, we define \( r_k(s, a) := -\nabla \hat{L}_k^{n,k}(\hat{\lambda}_k)(s, a) \). We then introduce the extended LP associated to \( \tilde{L}_k \) formulated over the variables \( q \) as

\[
\max_{q \in \Delta(\mathcal{S} \times \mathcal{A} \times \mathcal{S})} \sum_{s, a, s'} r_k(s, a)q(s, a, s')
\]

s.t. \( \forall j \in \mathcal{S} \),

\[
\sum_{a,s} q(j, a, s) - \sum_{a,s} q(s, a, j) = 0,
\]

\[
q(s, a, j) - (\hat{p}_k(j|s, a) + B_k(s, a, j)) \sum_{s'} q(s, a, s') \leq 0,
\]

\[
-q(s, a, j) + (\hat{p}_k(j|s, a) - B_k(s, a, j)) \sum_{s'} q(s, a, s') \leq 0,
\]

\[
\sum_{s'} q(s, a, s') \geq \eta.
\]

Crucially, the reparametrization in \( q \) enables to efficiently solve the problem as a “standard” LP. Let \( q_{k+1} \) be the solution of the problem above, then the state-action distribution can be easily recovered as

\[
\tilde{\phi}_{k+1}(s, a) := \sum_{s'} q_{k+1}(s, a, s'),
\]

and the corresponding policy is computed as in \( \tilde{\pi} \).

3.3 Sample Complexity

In order to derive the sample complexity of FW-ModeEst, we build on the analysis in [21] and show that the algorithm controls the regret w.r.t. the optimal solution of the problem \( \min \lambda \mathcal{L}_n(\lambda) \), i.e.,

\[
\mathcal{R}_n := \mathcal{L}_n(\bar{\lambda}_n) - \mathcal{L}_n(\lambda_{1,n}),
\]

where, similar to the previous section, \( \mathcal{L}_n \) can be either \( \mathcal{L}_n^\mathcal{E} \) or \( \mathcal{L}_n^\mathcal{W} \), \( \lambda_{1,n} \) is the corresponding optimal solution and \( \bar{\lambda}_n \) is the empirical frequency of visits after \( n \) steps.

We require \( \eta \) and the MDP to satisfy the two following assumptions:

4. This construction resembles the extended LP formulation used for loop-free stochastic shortest path problems in [18].

5. Refer to [21] for more details about the assumptions.

Assumption 1. The parameter \( \eta \) is such that

\[
\lambda_{1,n} \in \arg \min \mathcal{L}_n(\lambda), \quad \lambda_* \in \arg \min \mathcal{L}_\infty(\lambda),
\]

i.e., the optimal solutions in the restricted set \( \Lambda \) coincide with the unrestricted solutions for both \( \mathcal{L}_n \) and \( \mathcal{L}_\infty \).

Assumption 2. The MDP \( M \) is ergodic. We denote by \( \gamma_{\text{min}} := \min_{\pi} \gamma_{\pi}^* \), the smallest pseudo-spectral gap [16] over all stationary policies \( \pi \) and we assume \( \gamma_{\text{min}} > 0 \).

We are now ready to derive the model estimation error and sample complexity of FW-ModeEst. By extending the analysis from [21] to unknown transition model and integrating the regret analysis into our estimation error problem (see App. A), we obtain the following result.

Theorem 1. If FW-ModeEst is run with a budget \( n \) and the length of the episodes is set to \( r_k = 3k^2 - 3k + 1 \), then under Asm. 1 and 2 w.p. \( 1 - \delta \) and depending on the optimization function \( \mathcal{L} \) given as input, we have that

\[
\mathcal{E}_{\pi,n} = O \left( \frac{\mathcal{L}_\infty(\lambda_*^\mathcal{E})}{\sqrt{n}} + \frac{\Theta^\mathcal{E}}{n^{5/6}} \right),
\]

\[
\mathcal{W}_{\pi,n} = O \left( \frac{\mathcal{L}_\infty^W(\lambda_*^W) + \log(\mathcal{S}\mathcal{A})}{\sqrt{n}} + \frac{\Theta^W}{n^{5/6}} \right),
\]

where \( \Theta^\mathcal{E} \) (resp. \( \Theta^W \)) scales polynomially with MDP constants \( \mathcal{S}, \mathcal{A} \), and \( \gamma_{\text{min}}^{-1} \), and with algorithmic dependent constants such as \( \log(1/\delta) \) and \( \eta^{-1} \) when optimizing \( \mathcal{E} \) (resp. \( \mathcal{W} \)). From this result we immediately obtain the sample complexity of FW-ModeEst

\[
\mathcal{E}(\epsilon, \delta) = \Omega \left( \frac{\mathcal{L}_\infty(\lambda_*^\mathcal{E})^2}{\epsilon^2} + \frac{\Theta^\mathcal{E}}{\epsilon^{9/5}} \right),
\]

\[
\mathcal{W}(\epsilon, \delta) = \Omega \left( \frac{\mathcal{L}_\infty^W(\lambda_*^W) + \log(\mathcal{S}\mathcal{A})^2}{\epsilon^2} + \frac{\Theta^W}{\epsilon^{6/5}} \right).
\]

Model estimation performance. The previous result shows that FW-ModeEst successfully targets the model estimation problem with provable guarantees. In particular, the model estimation errors above display a leading error term scaling as \( O(n^{-1/2}) \) and a lower-order term scaling as \( O(n^{-5/6}) \). Under Asm. 2 any stochastic policy \( \pi \) with non-zero probability to execute any action is guaranteed to reach all states with a non-zero probability. As a result, the associated model estimation error would decrease as \( O(n^{-1/2}) \) as well, since \( T_{\pi,n}(s, a) \) grows linearly with \( n \) in all state-action pairs. Nonetheless, the linear growth could be very small as it depends on the smallest probability of reaching any state by following \( \pi \). As a result, despite the fact that \( \pi \) may achieve the same rate, FW-ModeEst actually performs much better. In fact, the term \( \mathcal{L}_\infty(\lambda_*)/n^{1/2} \) is an upper bound to the smallest error that can be achieved for the specific MDP.
We introduce a heuristic objective function for model estimation that allows more efficient learning with looser assumptions. To gain insight on how to optimize the weighted entropy function, with the weight of each state-action pair depending on its transitional noise.

**Definition 3.** For any non-negative weight function $w : S \times A \rightarrow \mathbb{R}^+$, the weighted entropy $H_w$ is defined on $\Delta(S \times A)$ as follows:

$$H_w(\lambda) := - \sum_{(s,a) \in S \times A} w(s,a) \lambda(s,a) \log(\lambda(s,a)).$$

To gain insight on $H_w$, we can argue that $w(s,a)$ represents the value or utility of each outcome $(s,a)$ (see e.g., [9]). Hence, the learner is encouraged to allocate importance on information regarding $(s,a)$ proportionally to the weight $w(s,a)$. This can be translated in biasing exploration towards region of the state-action space with high weights $w(s,a)$. Similar to [10], we also introduce a smoothed version of $H_w$ and show its properties.

**Lemma 2.** For any arbitrary set of non-negative weights $w$ — for which we define $W := \max_{s,a} w(s,a)$ — and for any smoothing parameter $\mu > 0$, we introduce the following smoothed proxy

$$H_{w,\mu}(\lambda) := \sum_{s,a} w(s,a) \lambda(s,a) \log \left( \frac{1}{\lambda(s,a) + \mu} \right).$$

The function $H_{w,\mu}$ satisfies the following properties:

1. $|\nabla H_{w,\mu}(\lambda)|_{s,a} = -w(s,a) \left( \log(\lambda(s,a) + \mu) + \frac{\lambda(s,a)}{\lambda(s,a) + \mu} \right)$.
2. For any $\mu \leq \frac{1}{e} - \frac{1}{SA}$, we have $|\nabla H_{w,\mu}(\lambda)|_{s,a} \geq 0$.
3. $H_{w,\mu}$ is concave in $\lambda$, as well as $W \log \left( \frac{1}{\mu} \right)$-Lipschitz continuous and $\frac{2W}{\mu}$-smooth.
4. We have $|H_w(\lambda) - H_{w,\mu}(\lambda)| \leq \mu SA W$.

In light of the intuition gained by inspecting the definition of $\mathcal{L}_n$, in the following we set the weights $w(s,a) = V(s,a) / \sqrt{SA}$, with $L := \log(SA/\delta)$, so as to encourage visiting state-action pairs with large transitional noise. We notice that, unlike the smooth versions of $\mathcal{L}$ and $W$, $H_{w,\mu}$ has a much smaller smoothness constant and its bias can be easily controlled by the choice of $\mu$. Moreover, the state-action distribution $\lambda$ does not appear in the denominator as in $\mathcal{L}_n$. All these factors suggest that learning how to optimize $H_{w,\mu}$ may be much simpler than directly targeting the model estimation error.

**4.2 Learning to Optimize the Weighted Entropy**

We seek to design a learning algorithm maximizing $H_w$. Since $H_{w,\mu}$ is concave, Lipschitz continuous and smooth
in the algorithm of [6] could have been readily applied to maximize it if the function was known. The key difference here is that the weights are unknown. We thus generalize the algorithm of [6] to handle this case. The resulting algorithm is outlined in Alg. 2. In it, $\text{EVI}(r, B, \varepsilon)$ denotes the standard extended value iteration scheme for reward function $r$, confidence region $B$ around the transition probabilities, and accuracy $\varepsilon$ (see [6] for more detail).

Following the terminology of [6], in our setting the vectorial outcome at any state-action pair $(s, a)$ is the standard basis vector for $(s, a)$ in $\mathbb{R}^{S \times A}$ (i.e., $1_{s, a}$). The objective is to minimize the regret

$$R_n^{\text{Ent}} := \min_{\lambda \in \Lambda^{(n)}} \mathcal{H}_w(\lambda) - \mathcal{H}_w(\tilde\lambda_t),$$

where $\Lambda^{(n)}$ is the space of state-action stationarity distributions and $\tilde\lambda_t$ is the empirical frequency of visits returned by the algorithm after $t$ steps. Notice that unlike in Sect. 2 and 3 the optimum is computed over $\lambda$ in the unrestricted set of stationary distribution (i.e., no $\eta$ lower bound), thus making this regret more general and challenging than [3].

We extend the scope of TOC-UCRL2 [6] to handle the objective function $\mathcal{H}_{w, \mu}$. For any time step $t$, let us define

$$\hat{\mathcal{H}}^+_t(\lambda) := \sum_{s, a} \frac{V^+_t(s, a)}{S^L} \lambda(s, a) \log\left(\frac{1}{\lambda(s, a) + \mu}\right).$$

Then we can derive a bound similar to Lem. 1:

**Lemma 3.** With probability at least $1 - \delta$, at any time step $t$, we have component-wise

$$\nabla \mathcal{H}(\tilde\lambda_t) \leq \nabla \hat{\mathcal{H}}^+_t(\tilde\lambda_t) \leq \nabla \mathcal{H}(\tilde\lambda_t) + u(t, \delta),$$

where the deviation $u(t, \delta)$ satisfies

$$[u(t, \delta)]_{s, a} = O\left(\sqrt{\text{log}(\frac{S A t}{\delta^3})}\right).$$

We now show how a simple extension of the algorithm of [6] can cope with the setting of unknown weights. At the beginning of each episode $k$, instead of feeding as reward the true, unknown current gradient, we simply feed an upper confidence bound of it, i.e., $\nabla \hat{\mathcal{H}}^+_k(\tilde\lambda_k)$, where the optimistic objective function $\hat{\mathcal{H}}^+_k$ — which depends on the smoothing parameter $\mu$ — is defined as in [10]. Initially, for small values of $t$, the weights are very similar, thus the algorithm targets a uniform exploration over the state-action space, that is, the original MAXENT objective. As more samples are collected, the estimation of the weights becomes more and more accurate, thus the exploration is gradually skewed towards regions of the state-action space with high transitional noise.

It is possible to extend the analysis of TOC-UCRL2 to this case and obtain a similar regret bound. As in [6], the only assumption we need is that the MDP $M$ is communicating, which is a much weaker requirement than Asm. 2.

**Assumption 3.** The MDP $M$ is communicating, with diameter $D := \max_{s \neq s'} \min_{\tau \in \mathcal{R}} \mathbb{E}[\tau(s \rightarrow s')] < +\infty$.

Then we can prove the following result.

**Theorem 2.** If Weighted-MAXENT is run with a budget $n$ and the smoothing parameter set to $\mu = \frac{1}{n^{1/3} S A \rho^2}$, then under Asm. 3 with overwhelming probability, we have

$$R_n^{\text{Ent}} = O\left(\frac{DS^{1/3}}{n^{1/3}} + \frac{D \sqrt{S A}}{\sqrt{n}}\right).$$

While they target different objectives, it is interesting to do a qualitative comparison between Thm. 2 and Thm. 1. We first extract from the error bound in Thm. 1 that FW-MODEST has a regret scaling as $\Theta/n^{1/3}$ w.r.t. $L^\circ(\lambda^\circ)$, While the rate is the same as in (11), the difference is that the constants are much better for the weighted MAXENT case. In fact, it scales linearly with the diameter $D$ and sublinearly with the size of the state-action space, instead of high-order polynomial dependencies on constants such as the constraint parameter $\eta$ and the MDP parameter $\gamma_{\min}$, which are likely to be very small in any practical application. As a result, we expect the weighted MAXENT version of TOC-UCRL2 to approach...
the performance of the optimal weighted entropy solution $\arg \min_\lambda \mathcal{H}_w(\lambda)$ faster than FW-ModEst approaches the performance of $\lambda^*$. While we do not have an explicit link between the model estimation error and the weighted entropy, we notice an important connection between $\mathcal{L}_w^\infty$ and $\mathcal{H}_w$ in the unconstrained case (i.e., when $\Lambda$ is reduced to the simplex over the state-action space with no constraint from the MDP). In this case, the solution to $\min_\lambda \mathcal{H}_w(\lambda)$ is $\lambda^*_w(\star, n)$, $\mathcal{L}_w^\infty(\star, n) \propto \exp(\alpha V(\star, n))$, with $\alpha$ a suitable constant, which shows that the optimal distribution has a direct connection with the transitional noise $V(\star, n)$. This exactly matches the intuition that a good distribution to minimize $\mathcal{L}_w^\infty$ should spend more time on states with larger noise. This connection is further confirmed in the numerical simulations we report in Sect. 5.

One may wonder why Alg. 2 cannot be applied to target the ModEst problem directly. In fact, Alg. 2 can be applied to any Lipschitz function (i.e., function with bounded gradient) and in the restricted set $\Lambda^p$, the function $\mathcal{L}_n$ used in FW-ModEst has indeed a bounded Lipschitz constant. Nonetheless, the regret analysis in Thm. 2 adapted from [6] requires evaluating the objective function at every step $t$, which in our case means evaluating $\mathcal{L}_n$ at the empirical frequency $\hat{\lambda}_t$. Unfortunately, $\hat{\lambda}_t$ is a random quantity and it may not belong $\Lambda^p$ at each step. This justifies the ergodicity assumption and the per-episode analysis of FW-ModEst, where episodes are long enough so that the ergodicity of the MDP guarantees that each state-action pair is visited enough and $\mathcal{L}_n$ is evaluated only in the restricted set, where it is well-behaved.

5 NUMERICAL SIMULATIONS

The objective of the section is to illustrate how Weighted-MaxEnt and FW-ModEst are able to effectively adapt to the characteristics of the environment. We consider two domains (NoisyRiverSwim and Wheel-of-Fortune) with high level of stochasticity, i.e., $V := \sum_{s,a} V(s, a)/(s,a)$ is large, and the transitional noise is heterogeneously spread across the state-action space, i.e., $\sigma(V) := \sqrt{\frac{1}{s,a} \sum_{s,a} (V(s, a) - V)^2}$ is large. A complete description of the environments is provided in App. [C].

Optimal solution. We start evaluating the error $\mathcal{E}_{\pi^*, n}$ associated to the optimal distribution $\lambda^*$, computed for MaxEnt, Weighted-MaxEnt and FW-ModEst. For each state-action pair $(s, a)$ and algorithm $\mathfrak{I}$, we estimate the transition kernel $p^\mathfrak{I}_{\pi^*, n}(s' | s, a)$ by generating $k = n \cdot \lambda^*_w(s, a) \cdot 1$ samples from $p(\cdot | s, a)$. We select $n = 2 \cdot 10^6$ to simulate the asymptotic behavior of the algorithms. As shown in Tab. 1, Weighted-MaxEnt and FW-ModEst recover almost identical policies in these domains, leading to very similar estimation errors. As expected, MaxEnt is outperformed by the proposed algorithms at the task of model estimation. Refer to App. [C] for the values of $\lambda^*$.

Learning. Now that we have shown that Weighted-MaxEnt and FW-ModEst have a similar asymptotic performance, we can focus on the learning process. We start considering the NoisyRiverSwim(12) problem. Fig. 1 (left) shows that MaxEnt and uniform policy have a similar behavior in this domain. Both approaches are outperformed by FW-ModEst and Weighted-MaxEnt. Despite having the same optimal solution, Weighted-MaxEnt performs better than FW-ModEst. As suggested by the qualitative comparison between Thm. 1 and Thm. 2, the gap is due to the different learning process. In particular, we found Weighted-MaxEnt to be more numerically stable and more sample efficient than FW-ModEst. We have observed a similar behavior in the Wheel-of-Fortune environment (see Fig. 1 (right)). In all our experiments, Weighted-MaxEnt has outperformed the other algorithms. We refer the reader to App. [C] for more details and additional experiments on randomly generated MDPs.

6 CONCLUSION

We studied the problem of reward-free exploration for model estimation and designed FW-ModEst, the first algorithm for this problem with sample complexity guarantees. We also introduced a heuristic algorithm (Weighted-MaxEnt) which requires much less restrictive assumptions and achieves better performance than FW-ModEst and MaxEnt in our numerical simulations.
References

[1] M. Araya-López, O. Buffet, V. Thomas, and F. Charpillet. Active learning of mdp models. In European Workshop on Reinforcement Learning, pages 42–53. Springer, 2011.

[2] J.-Y. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In International conference on algorithmic learning theory, pages 150–165. Springer, 2007.

[3] A. Beck. First-order methods in optimization, volume 25. SIAM, 2017.

[4] Q. Berthet and V. Perchet. Fast rates for bandit optimization with upper-confidence frank-wolfe. In Advances in Neural Information Processing Systems, pages 2225–2234, 2017.

[5] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor–critic algorithms. Automatica, 45(11):2471–2482, 2009.

[6] W. C. Cheung. Exploration-exploitation trade-off in reinforcement learning on online markov decision processes with global concave rewards. arXiv preprint arXiv:1905.06466, 2019.

[7] W. C. Cheung. Regret minimization for reinforcement learning with vectorial feedback and complex objectives. In Advances in Neural Information Processing Systems, pages 724–734, 2019.

[8] R. Fruit, M. Pirotta, and A. Lazaric. Improved analysis of UCRL2B. 2019. URL https://rlgammazero.github.io/docs/ucrl2bImproved.pdf.

[9] S. Guiasu. Weighted entropy. Reports on Mathematical Physics, 2(3):165–179, 1971.

[10] E. Hazan, S. Kakade, K. Singh, and A. Van Soest. Provably efficient maximum entropy exploration. In International Conference on Machine Learning, pages 2681–2691, 2019.

[11] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th international conference on machine learning, number CONF, pages 427–435, 2013.

[12] C. Jin, A. Krishnamurthy, M. Simchowitz, and T. Yu. Reward-free exploration for reinforcement learning. arXiv preprint arXiv:2002.02794, 2020.

[13] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. Machine learning, 49(2-3):209–232, 2002.

[14] S. Lim and P. Auer. Autonomous exploration for navigating in MDPs. In Proceedings of the 25th Annual Conference on Learning Theory, 2012.

[15] A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. arXiv preprint arXiv:0907.3740, 2009.

[16] D. Paulin. Concentration inequalities for markov chains by marton couplings and spectral methods. Electronic Journal of Probability, 20, 2015.

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

[18] A. Rosenberg and Y. Mansour. Online convex optimization in adversarial markov decision processes. In International Conference on Machine Learning, pages 5478–5486, 2019.

[19] S. Shekhar, M. Ghavamzadeh, and T. Javidi. Adaptive sampling for estimating multiple probability distributions. arXiv preprint arXiv:1910.12406, 2019.

[20] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for markov decision processes. Journal of Computer and System Sciences, 74(8):1309–1331, 2008.

[21] J. Tarbouriech and A. Lazaric. Active exploration in markov decision processes. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 974–982, 2019.
A Proofs in Sect. 3

Proof of Prop. 2. By direct application of Bernstein’s inequality, we have that for any triplet \((s, a, s') \in S \times A \times S\), with probability at least \(1 - \delta/3\),

\[
|p(s'|s,a) - \hat{p}_{\pi,n}(s'|s,a)| \leq \sqrt{\frac{2p(s'|s,a)(1-p(s'|s,a))\log(6/\delta)}{T_{\pi,n}(s,a)}} + \frac{\log(6/\delta)}{3T_{\pi,n}(s,a)}.
\]

As a result, we can now derive a direct upper bound on the following function:

\[
\max_{s,a} \frac{V(s,a)\sqrt{T_{\pi,n}(s,a)}}{T_{\pi,n}(s,a)}
\]

where (a) stems from the fact that under \(\pi\), \(\sigma_{\pi}(s,a) \leq n\), (b) stems from the fact that \(\max \{x, y\} = \frac{x+y+|x-y|}{2} \geq \frac{x+y}{2}\), and (c) introduces the following function:

\[
F(s,a;T) := \frac{V(s,a)}{\sqrt{T(s,a)} + 1} + \frac{S}{T(s,a) + 1}.
\]

In a similar way, we have that

\[
W_{\pi,n} \leq 4\log \left(\frac{6SA_{n}}{\delta}\right) \max_{s,a} F(s,a;T_{\pi,n}).
\]

Proof of Lemma 7. From [15], with probability at least \(1 - \delta\), for any \((s, a, s') \in S \times A \times S\) and any time step \(t \geq 1\), we have the following concentration inequality for the estimation of the standard deviation

\[
\left|\sqrt{\sigma^{2}_{\pi}(s'|s,a)} - \hat{\sigma}^{2}_{\pi}(s'|s,a)\right| \leq \sqrt{\log \left(\frac{4S^{2}(T_{\pi,n}(s,a))^{2}}{\delta}\right)}.
\]

As a result, summing over the next states \(s' \in S\) gives the first statement of Lem. 1. As for second statement, from Thm. 10 of [8], the fact that the confidence intervals are constructed using the empirical Bernstein inequality (15) implies that, with high probability, the true transition model belongs to the confidence set \(B_{t}\) at any time \(t\).

Proof of Thm. 7. A) First, we extend the regret bound of [21] to handle the setting of an unknown transition model. For clarity, we adopt the same notation as in [21] and unravel the analysis. We write \(L := L_{n}\). Denote by \(\rho_{k+1}\) the approximation error at the end of episode \(k\), i.e., \(\rho_{k+1} := L(\lambda_{k+1}) - L(\lambda^{*})\). Then from Eq. (24) of App. D.2 of [21],

\[
\rho_{k+1} = (1 - \beta_{k})\rho_{k} + C_{n}\beta_{k}^{2} + \beta_{k}\epsilon_{k+1} + \beta_{k}\Delta_{k+1},
\]

where \(\beta_{k} := \tau_{k}/(t_{k+1} - 1)\) and

\[
\epsilon_{k+1} := \langle \nabla L_{n}(\lambda_{k}), \hat{\psi}_{k+1} - \psi_{k+1} \rangle, \quad \Delta_{k+1} := \langle \nabla L_{n}(\lambda_{k}), \tilde{\psi}_{k+1} - \hat{\psi}_{k+1} \rangle.
\]

The tracking error \(\Delta_{k+1}\) can be bounded exactly as in [21]. As for the optimization error \(\epsilon_{k+1}\), we first set

\[
x_{k+1} := \sum_{s,a} \hat{\psi}_{k+1}^{+}(s,a) \frac{\alpha(t_{k} - 1, s, a, \delta)}{\lambda_{k}(s,a)^{2}},
\]

where from Lem. 1 we have

\[
\alpha(t_{k} - 1, s, a, \delta) := 2S \frac{\log(2S^{2}AT_{\pi,n}(s,a))}{T_{\pi,n}(s,a)}.
\]

Denote by \(Q\) the event under which Lem. 1 holds. We get under \(Q\) that

\[
\langle \nabla L(\lambda_{k}), \hat{\psi}_{k+1}^{-} \rangle \leq \langle \nabla L^{+}(\lambda_{k}), \hat{\psi}_{k+1}^{-} \rangle + 2x_{k+1}
\]

\[
= \min_{\rho \in B_{k}, \lambda \in \Lambda^{(p)}} \langle \nabla L^{+}(\lambda_{k}), \lambda \rangle + 2x_{k+1}
\]

\[
\leq \langle \nabla L^{+}(\lambda_{k}), \psi_{k+1}^{*} \rangle + 2x_{k+1}
\]

\[
\leq \langle \nabla L(\lambda_{k}), \psi_{k+1}^{*} \rangle + 2x_{k+1},
\]

where (a) stems from the fact that under \(Q\) we have \(p \in B_{k}\) (Lem. 1) and that \(\psi_{k+1}^{*} \in \Lambda^{(p)}\). Additionally, the error \(x_{k+1}\) can be bounded exactly as in [21]. This enables to control the optimization error \(\epsilon_{k+1}\) in a similar manner. All in all, the fact that the transition dynamics are unknown does not affect the final regret bound of Thm. 1 of [21].

B) We now proceed in deriving a sample complexity result for \(\hat{E}_{\pi,n}\). We can bound the regret \(R_{n}^{E}\), which is defined as follows

\[
R_{n}^{E} := L_{n}(\lambda_{n}) - L_{n}(\lambda^{E}_{1,n}),
\]
where \( \lambda^x_{\pi,n} \in \arg \min_{\lambda \in \Lambda_n} \mathcal{L}^x_n \).

From Thm. 1 of [21], if we select episode lengths \( \tau_k \) scaling as \( 3k^2 - 3k + 1 \), we have that with overwhelming probability

\[
\mathcal{R}^\epsilon_n = \tilde{O}\left( \frac{\Theta^\epsilon}{n^{1/3}} \right),
\]

(12)

with \( \Theta^\epsilon \) scaling polynomially with MDP constants \( S, A \), and \( \gamma_{\min}^{-1} \), and with algorithmic dependent constants such as \( \log(1/\delta) \) and \( \eta^{-1} \) (see the proof of Thm. 1 of [21] for the exact dependencies). Using that \( \lambda^x_\pi \in \Lambda_n \), we have

\[
\mathcal{L}^x_n(\lambda^x_\pi) \leq \mathcal{L}^x_n(\lambda^x_{\pi,n}) + \frac{S}{\eta \sqrt{n}} + \tilde{O}\left( \frac{\Theta^\epsilon}{n^{1/3}} \right).
\]

(13)

Combining Eq. (12) and (13) yields

\[
\epsilon_{\pi,n} \leq 4 \log\left( \frac{6SAn}{\delta} \right) \left( \mathcal{L}^x_n(\lambda^x_{\pi,n}) + \frac{S}{\eta \sqrt{n}} + \tilde{O}\left( \frac{\Theta^\epsilon}{n^{1/3}} \right) \right).
\]

Consequently, we get

\[
\epsilon_{\pi,n} = \tilde{O}\left( \frac{\mathcal{L}^x_n(\lambda^x_{\pi,n})}{\sqrt{n}} + \frac{\Theta^\epsilon}{n^{1/3}} \right).
\]

C) Finally, we derive a sample complexity result for \( W_{\pi,n} \). The function \( \mathcal{L}^W_n \) that is fed to the learning algorithm as objective function is both convex and smooth in \( \pi \) on the restricted simplex \( \Lambda_n \) (see Prop. [5]). Moreover, at the beginning of each episode \( k \), we can compute an optimistic estimate of the unknown gradient by replacing the unknown \( V(s,a) \) with the optimistic quantities \( \tilde{V}^+_k(s,a) \). Let us define \( \lambda^W_{\pi,n} \in \arg \min_{\lambda \in \Lambda_n} \mathcal{L}^W_n \). The regret is defined as follows

\[
\mathcal{R}^W_n := \mathcal{L}^W_n(\lambda^W_n) - \mathcal{L}^W_n(\lambda^W_{\pi,n}),
\]

Then we have that

\[
\mathcal{R}^W_n = \tilde{O}\left( \Theta^W_{\pi,n} \right),
\]

with \( \Theta^W \) scaling polynomially with MDP constants \( S, A \), and \( \gamma_{\min}^{-1} \), and with algorithmic dependent constants such as \( \log(1/\delta) \) and \( \eta^{-1} \). As a result, we have

\[
\mathcal{L}^W_n(\lambda^W_n) \leq \mathcal{L}^W_n(\lambda^W_{\pi,n}) + \mathcal{R}^W_n
\]

\[
\leq \mathcal{L}^W_n(\lambda^W_{\pi,n}) + \mathcal{R}^W_n
\]

\[
\leq \mathcal{L}^W_n(\lambda^W_{\pi,n}) + \mathcal{R}^W_n
\]

\[
\leq \mathcal{L}^W_n(\lambda^W_{\pi,n}) + \mathcal{R}^W_n
\]

\[
\leq \mathcal{L}^W_n(\lambda^W_{\pi,n}) + \mathcal{R}^W_n
\]

where the chain of inequalities follows from the definitions of \( \lambda^W \) and \( \lambda^W_{\pi,n} \) as well as point 2) of Prop. [4]. This finally yields

\[
W_{\pi,n} = \tilde{O}\left( \frac{\mathcal{L}^W_n(\lambda^W_n) + \log(SA) + \Theta^W_{\pi,n}}{n^{1/3}} \right).
\]

\[\square\]

B) Proofs in Sect. 4

Proof of Lemma [2]. The inequalities stem from a combination of Lem. [4] and Lem. [1]. More specifically, the first inequality comes from point 2. of Lem. [2] and from Lem. [1], which states that \( V(s,a) \leq V^+_k(s,a) \). As for the second inequality, we notice that

\[
\| \nabla H^- \tilde{\lambda}_n \| - \nabla H(\tilde{\lambda}_n) \| \leq \frac{V^+_k(s,a) - V(s,a)}{\sqrt{SA}} \left( \log \left( \frac{1}{\mu} \right) \right)
\]

\[
= O\left( \frac{\log(SA)}{T_k(s,a)} \right).
\]

\[\square\]

Proof of Lemma [2]. Properties 1., 3. and 4. stem from Lem. 4.3 of [10]. As for property 2., we notice that by choice of \( \mu \), we have for any \( (s,a) \) that \( \mu \leq \frac{1}{\varepsilon} - \lambda(s,a) \), and hence

\[
\frac{1}{\log(\lambda(s,a) + \mu)} \geq 1 \geq \frac{\lambda(s,a)}{\lambda(s,a) + \mu},
\]

so we finally get 2. since the weights are non-negative.

\[\square\]

Proof of Thm. [2]. We unravel the analysis of [6], whose Eq. (11) yields

\[
R_n^\text{Ent} \leq \frac{4 \log \left( \frac{1}{\mu} \right) \log(n)}{n} + \frac{1}{n} \sum_{i=1}^n \nabla H(\tilde{\lambda}_n)^\top (\lambda^* - \mathbb{1}_{s,a}),
\]

since in our setting the vectorial outcome at any state-action pair \( (s,a) \) considered by [6] corresponds to the standard basis vector for \( (s,a) \) in \( \mathbb{R}^{S \times A} \) (i.e., \( \mathbb{1}_{s,a} \)). We
The term \( Z \) can decompose the second term as follows:

\[
\sum_{t=1}^{n} \nabla H(\tilde{\lambda}_t) (\lambda^* - \mathbb{1}_{s_t,a_t}) \\
= \sum_{t=1}^{n} \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t) (\lambda^* - \mathbb{1}_{s_t,a_t}) \\
+ \sum_{t=1}^{n} \left( \nabla H(\tilde{\lambda}_t) - \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t) \right) (\lambda^* - \mathbb{1}_{s_t,a_t}) \\
:= Z_t
\]

On the one hand, Lem. 3 yields that with high probability, \( \nabla H(\tilde{\lambda}_t) - \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t) \leq 0 \) component-wise. On the other hand, \( \lambda^*(s,a) - \mathbb{1}_{s_t,a_t} \) belongs to \([-1, 0]\) only at \((s, a) = (s_{t-1}, a_{t-1})\), and is positive otherwise. Hence, we have:

\[
Z_t \leq \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t)_{s_t,a_t} - [\nabla H(\tilde{\lambda}_t)]_{s_t,a_t} \leq [u(t, \delta)]_{s_t,a_t},
\]

where the deviation \( u(t, \delta) \) is controlled by Lem. 3, which entails that:

\[
\frac{1}{n} \sum_{t=1}^{n} Z_t = O\left( \sqrt{\frac{\Gamma S A \log \left( \frac{n}{\delta^4} \right)}{n}} \right).
\]

The term \( \frac{1}{n} \sum_{t=1}^{n} \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t) (\lambda^* - \mathbb{1}_{s_t,a_t}) \) can be bounded following the analysis of [6], since \( \nabla \tilde{H}_{k,t}^+(\tilde{\lambda}_t) \) corresponds exactly to the rewards fed to the EVI procedure at the beginning of episode \( k \). Moreover, from Lem. 1 we have:

\[
\max_{s,a,t} \frac{\hat{V}_{t}^+(s,a)}{\sqrt{SL}} \leq 2.
\]

Hence from point 3. of Lem. 2 we get that the Lipschitz-constant of \( H_{k,t}^+ \) at any time step \( t \) is upper bounded by \( Q := 2 \log \left( \frac{1}{p} \right) \). As a result, by setting the gradient threshold as equal to \( Q \) in Alg. 2, we obtain the same regret bound as Eq. (6.3) of [6]. We finally thus get:

\[
R_{En^t} = O\left( DS^{1/3} \frac{1}{n^{1/3}} + D \sqrt{\frac{\Gamma S A}{\sqrt{n}}} \right).
\]

### C. Numerical Simulations

#### C.1 Description of the environments used

**NoisyRiverSwim.** NoisyRiverSwim is a noisy variant of the standard environment RiverSwim [20] (see Fig. 2). At each of the \( S \) states forming the chain, the two usual actions are available, plus two additional ones: action \( a_{even} \) (resp. action \( a_{odd} \)) brings any even (resp. odd) state to any other state of the environment. In odd states, action \( a_{even} \) self-loops deterministically, and vice versa. This variant of RiverSwim injects some additional stochasticity in the transition dynamics, over which the agent has some control insofar as the behavior of the “random teleportation” at a given state only takes place only if the correct action is taken (i.e., \( a_{even} \) if the state is even, \( a_{odd} \) otherwise).

**Wheel-of-Fortune.** This environment consists of \( S \) states (see Fig. 3), one of which is the center of the wheel \( s_0 \) and the remaining \( S - 1 \) form the ring of the wheel \( \{s_1, s_2, \ldots, s_{S-1}\} \). \( A = 5 \) actions are available at each state:

- In the center state \( s_0 \), \( p(s_0|s_0, a) = 1 \) for all actions except for one action SPIN \( a^\dagger \) for which \( \forall n \in \{1, \ldots, S-1\}, p(s_n|s_0, a^\dagger) = 1/(S-1) \).

- For all remaining states, i.e., \( \{s_1, \ldots, s_{S-1}\} \), the actions LEFT, RIGHT, SELF-LOOP, CENTER bring deterministically the agent to the left neighboring state, the right neighboring state, the same state and the center state, respectively. The action NOISY equiprobably yields the outcome of either of the 4 previous actions.

**Garnet.** We also consider standard randomly generated environments [3]. A Garnet instance \( G(S, A, b) \) has \( S \) states and \( A \) actions, and for each state-action pair its branching factor \( \Gamma(s,a) \) is randomly sampled in \([0, b] \).
Figure 3: Example of the Wheel-Of-Fortune environment with \( S = 7 \) states.

C.2 Optimal solution

In Sect. 5, we reported the estimation error \( E_{\pi^*, n} \), computed using the optimal solution \( \lambda^{\pi^*, n}_A \) and \( n = 2 \cdot 10^6 \) samples. As shown in Tab. 1, WEIGHTED-MAXENT and FW-MODEST have almost identical estimation error. This is due to the fact that their optimal solutions are similar.

Consider the Wheel(5) environment, with \( \eta = 0.0001 \) and \( \mu = 0 \). The optimal solutions are

\[
\begin{align*}
\lambda^{\text{WEIGHTED-MAXENT}}_* &= \begin{pmatrix}
0.0 & 0.0 & 0.0 & 0.0 & 0.2 \\
0.0 & 0.0 & 0.0 & 0.2 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.2 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.2 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.2 & 0.0
\end{pmatrix}, \\
\lambda^{\text{FW-MODEST}}_* &= \begin{pmatrix}
\eta & \eta & \eta & \eta & 0.1999 \\
\eta & \eta & \eta & \eta & 0.1995 \\
\eta & \eta & \eta & \eta & 0.1995 \\
\eta & \eta & \eta & \eta & 0.1995 \\
\eta & \eta & \eta & \eta & 0.1995
\end{pmatrix}, \\
\lambda^{\text{MAXENT}}_* &= \begin{pmatrix}
0.043 & 0.043 & 0.043 & 0.043 & 0.1048 \\
0.043 & 0.043 & 0.043 & 0.0176 & 0.0344 \\
0.043 & 0.043 & 0.043 & 0.0176 & 0.0344 \\
0.043 & 0.043 & 0.043 & 0.0176 & 0.0344 \\
0.043 & 0.043 & 0.043 & 0.0176 & 0.0344
\end{pmatrix}.
\end{align*}
\]

It is easy to notice that WEIGHTED-MAXENT and FW-MODEST have almost identical solution, the only difference being due to the lower bound \( \eta \). We observed similar results in the environment NoisyRiverSwim(12).

C.3 Additional experiments

We first performed an additional experiment on the Wheel-Of-Fortune environment with 10 and 20 states. In this case as well, WEIGHTED-MAXENT outperforms all the other algorithms, as shown by see Fig. 4. Finally, we performed multiple experiments on the Garnet domain. Fig. 5 reports the average and worst-case model estimation errors \( (\mathcal{E}, \mathcal{W}) \) for each randomly generated environment after \( n \) steps. The experimental protocol is as follows: we generated 10 random instances of the Garnet domain and for each instance we ran 20 times each algorithm in order to report the mean and standard deviation of the errors. In almost all the experiments, the best performance is achieved by WEIGHTED-MAXENT (see Fig. 5). We observe that, as expected, the gap in performance with MAXENT increases with the stochasticity of the environment, which is captured by the quantity \( \sigma(V) \).
Figure 4: Estimation error $\mathcal{E}_{\pi, n}$ for the environment Wheel(10) and Wheel(20) (averaged over 20 runs).

First table: 10 randomly generated instances of $G(5, 5, 5)$, with $R = 20$ runs and a budget of $n = 10000$:

| $G(5,5,5)$ | Uniform | MAXENT | WEIGHTED-MAXENT |
|------------|---------|--------|-----------------|
| $0$        | $0.306$ $0.3058 \pm 0.0026$ | $0.1249 \pm 0.03$ | $0.112 \pm 0.0093$ |
| $1$        | $0.274$ $0.267 \pm 0.0026$ | $0.1348 \pm 0.0017$ | $0.1205 \pm 0.0067$ |
| $2$        | $0.234$ $0.18 \pm 0.0021$ | $0.1209 \pm 0.0074$ | $0.1182 \pm 0.0134$ |
| $3$        | $0.222$ $0.1384 \pm 0.0023$ | $0.1269 \pm 0.0175$ | $0.1087 \pm 0.0062$ |
| $4$        | $0.226$ $0.1555 \pm 0.0022$ | $0.1133 \pm 0.0091$ | $0.1085 \pm 0.0103$ |
| $5$        | $0.116$ $0.0397 \pm 0.0023$ | $0.1155 \pm 0.0112$ | $0.1097 \pm 0.0094$ |
| $6$        | $0.2555$ $0.0351 \pm 0.0018$ | $0.1272 \pm 0.0109$ | $0.1166 \pm 0.0076$ |
| $7$        | $0.2543$ $0.0299 \pm 0.0019$ | $0.1338 \pm 0.0147$ | $0.1207 \pm 0.0124$ |
| $8$        | $0.2097$ $0.0413 \pm 0.0023$ | $0.1358 \pm 0.0215$ | $0.141 \pm 0.002$ |
| $9$        | $0.2534$ $0.0383 \pm 0.0020$ | $0.1142 \pm 0.0010$ | $0.1185 \pm 0.0013$ |

Second table: 10 randomly generated instances of $G(10, 10, 5)$, with $R = 20$ runs and a budget of $n = 20000$:

| $G(10,10,5)$ | Uniform | MAXENT | WEIGHTED-MAXENT |
|-------------|---------|--------|-----------------|
| $0$         | $0.1711$ $0.0389 \pm 0.0018$ | $0.1417 \pm 0.0098$ | $0.1398 \pm 0.0115$ |
| $1$         | $0.1806$ $0.0332 \pm 0.0014$ | $0.1266 \pm 0.0082$ | $0.1227 \pm 0.0111$ |
| $2$         | $0.1591$ $0.0434 \pm 0.0013$ | $0.1436 \pm 0.0157$ | $0.1386 \pm 0.0087$ |
| $3$         | $0.1986$ $0.0333 \pm 0.0015$ | $0.1367 \pm 0.0082$ | $0.1271 \pm 0.0113$ |
| $4$         | $0.1843$ $0.0378 \pm 0.0017$ | $0.1351 \pm 0.0125$ | $0.1209 \pm 0.0075$ |
| $5$         | $0.1794$ $0.0358 \pm 0.0016$ | $0.1267 \pm 0.0086$ | $0.1195 \pm 0.0096$ |
| $6$         | $0.1857$ $0.0338 \pm 0.0011$ | $0.1172 \pm 0.0142$ | $0.1211 \pm 0.0111$ |
| $7$         | $0.1703$ $0.0393 \pm 0.0013$ | $0.1016 \pm 0.0062$ | $0.1153 \pm 0.0129$ |
| $8$         | $0.1833$ $0.0318 \pm 0.0012$ | $0.1054 \pm 0.0111$ | $0.1097 \pm 0.0116$ |
| $9$         | $0.1992$ $0.0332 \pm 0.0015$ | $0.1282 \pm 0.0069$ | $0.1266 \pm 0.0098$ |

Third table: 10 randomly generated instances of $G(20, 10, 5)$, with $R = 20$ runs and a budget of $n = 40000$:

| $G(20,10,5)$ | Uniform | MAXENT | WEIGHTED-MAXENT |
|--------------|---------|--------|-----------------|
| $0$          | $0.1168$ $0.0399 \pm 0.0012$ | $0.1903 \pm 0.0193$ | $0.1558 \pm 0.0141$ |
| $1$          | $0.1403$ $0.0343 \pm 0.0011$ | $0.1878 \pm 0.0253$ | $0.1388 \pm 0.0095$ |
| $2$          | $0.1294$ $0.0391 \pm 0.0009$ | $0.1751 \pm 0.0114$ | $0.1431 \pm 0.0084$ |
| $3$          | $0.1334$ $0.0377 \pm 0.001$ | $0.1644 \pm 0.0118$ | $0.1593 \pm 0.0067$ |
| $4$          | $0.1307$ $0.0321 \pm 0.0009$ | $0.1322 \pm 0.0118$ | $0.1445 \pm 0.0107$ |
| $5$          | $0.1272$ $0.0387 \pm 0.0011$ | $0.1706 \pm 0.0159$ | $0.1599 \pm 0.0169$ |
| $6$          | $0.1257$ $0.0337 \pm 0.0014$ | $0.1615 \pm 0.0188$ | $0.1388 \pm 0.0088$ |
| $7$          | $0.1261$ $0.0363 \pm 0.001$ | $0.1533 \pm 0.0114$ | $0.1448 \pm 0.0111$ |
| $8$          | $0.1306$ $0.0348 \pm 0.0009$ | $0.1407 \pm 0.0079$ | $0.1496 \pm 0.0106$ |
| $9$          | $0.1275$ $0.0335 \pm 0.0011$ | $0.1514 \pm 0.0010$ | $0.1397 \pm 0.0086$ |

Figure 5: Experiments on the Garnet environment. The first column is the index of the randomly generated Garnet $G(S, A, b)$, the second column is the $\sigma(V)$ of the corresponding Garnet instance. For each algorithm (uniform policy, MAXENT and WEIGHTED-MAXENT), the first column denotes the average estimation error $\mathcal{E}$ after $n$ steps, while the second column denotes the worst-case estimation error $\mathcal{W}$ after $n$ steps. The value after the $\pm$ sign denotes the standard deviation of the errors over the $R = 20$ runs.