Kinematic State Abstraction and Provably Efficient Rich-Observation Reinforcement Learning

Dipendra Misra [1] **Mikael Henaff** [1] **Akshay Krishnamurthy** [1] **John Langford** [1]

Abstract

We present an algorithm, HOMER, for exploration and reinforcement learning in rich observation environments that are summarizable by an unknown latent state space. The algorithm interleaves representation learning to identify a new notion of *kinematic state abstraction* with strategic exploration to reach new states using the learned abstraction. The algorithm provably explores the environment with sample complexity polynomial in the number of latent states and time horizon. Crucially, the observation space could be infinitely large. This guarantee enables efficient global policy optimization for any reward function. On the computational side, we show that HOMER can be implemented efficiently whenever certain supervised learning problems are tractable. Empirically, we evaluate HOMER on a challenging exploration problem, where we show that the algorithm is exponentially more sample efficient than standard reinforcement learning baselines.

1. Introduction

Modern reinforcement learning applications call for agents that operate directly from rich sensory information such as megapixel camera images. This rich information enables representation of detailed, high-quality policies and obviates the need for hand-engineered features. However, exploration in such settings is notoriously difficult and, in fact, statistically intractable in general (Jaksch et al., 2010; Lattimore & Hutter, 2012; Krishnamurthy et al., 2016). Despite this, many environments are highly structured and do admit sample efficient algorithms (Jiang et al., 2017); indeed, we may be able to summarize the environment with a simple state space and extract these states from raw observations. With such structure, we can leverage techniques from the well-studied tabular setting to explore the environment (Hazan et al., 2018), efficiently recover the underlying dynamics (Strehl & Littman, 2008), and optimize any reward function (Kearns & Singh, 2002; Brafman & Tennenholtz, 2002; Strehl et al., 2006; Dann et al., 2017; Azar et al., 2017; Jin et al., 2018). But can we learn to decode a simpler state from raw observations alone?

The main difficulty is that learning a state decoder, or a compact representation, is intrinsically coupled with exploration. On one hand, we cannot learn a high-quality decoder without gathering comprehensive information from the environment, which may require a sophisticated exploration strategy. On the other hand, we cannot tractably explore the environment without an accurate decoder. These interlocking problems constitute a central challenge in reinforcement learning, and a provably effective solution remains elusive despite decades of research (Mccallum, 1996; Ravindran, 2004; Jong & Stone, 2005; Li et al., 2006; Bellemare et al., 2016; Nachum et al., 2019).

In this paper, we provide a solution for a significant subclass of problems known as Block Markov Decision Processes (MDPs) (Du et al., 2019), in which the agent operates directly on rich observations that are generated from a small number of unobserved latent states. Our algorithm, HOMER, learns a new reward-free state abstraction called *kinematic inseparability*, which it uses to drive exploration of the environment. Informally, kinematic inseparability aggregates observations that have the same forward and backward dynamics. When observations have shared backward dynamics, a single policy simultaneously maximizes the probability of reaching them, which is useful for exploration. Shared forward dynamics is naturally useful for recovering the latent state space and model. Most importantly, we show that kinematic inseparability can be recovered from a bottleneck in a regressor trained on a contrastive estimation problem derived from raw observations.

HOMER performs strategic exploration by training policies to visit each kinematically inseparable abstract state, resulting in a *policy cover*. These policies are constructed via a reduction to contextual bandits (Bagnell et al., 2004), using a synthetic reward function that incentivizes reaching an abstract state. Crucially, HOMER interleaves learning the state

[*]Equal contribution [1]Microsoft Research, New York, NY. Correspondence to: Dipendra Misra <dimisra@microsoft.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

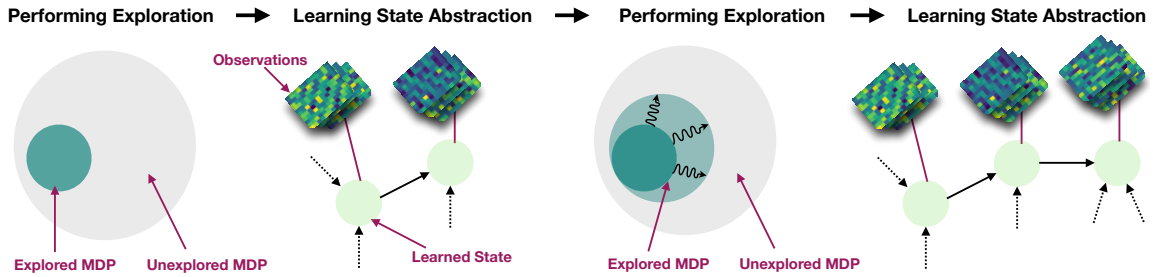


Figure 1: HOMER learns a set of exploration policies and a state abstraction function by iterating between exploring using the current state abstraction and refining the state abstraction based on the new experience.

abstraction and policy cover in an inductive manner: we use the policies from a coarse abstraction to reach new states, which enables us to refine the state abstraction and learn new policies (See Figure 1 for a schematic). Each process is essential to the other. Once the policy cover is constructed, we can use it to efficiently gather the information necessary to find a near-optimal policy for any reward function.

We analyze the statistical and computational properties of HOMER in episodic Block MDPs. We prove that HOMER learns to visit every latent state and also learns a near-optimal policy for any given reward function with a number of trajectories that is polynomial in the number of latent states, actions, horizon, and the complexity of two function classes used by the algorithm. There is no explicit dependence on the observation space size. The main assumptions are that the latent states are reachable and that the function classes are sufficiently expressive. There are no identifiability or determinism assumptions beyond decodability of the Block MDP, resulting in significantly greater scope than prior work (Du et al., 2019; Dann et al., 2018). On the computational side, HOMER operates in a reductions model and can be implemented efficiently whenever certain supervised learning problems are tractable.

Empirically, we evaluate HOMER on a challenging reinforcement learning problem with high-dimensional observations, precarious dynamics, and sparse, misleading rewards. The problem is googol-sparse: the probability of encountering an optimal reward through random search is 10^{-100}. HOMER recovers the underlying state abstraction for this problem and consistently finds a near-optimal policy, outperforming popular baselines that use naive exploration strategies (Mnih et al., 2016; Schulman et al., 2017) or more sophisticated exploration bonuses (Burda et al., 2019), as well as the recent PAC-RL algorithm of Du et al. (2019).

2. Preliminaries

We consider reinforcement learning (RL) in episodic Block Markov Decision Processes (Block MDP), first introduced by Du et al. (2019). A Block MDP $\mathcal{M}$ is described by a large (possibly infinite) observation space $\mathcal{X}$, a finite latent unobserved state space $\mathcal{S}$, a finite set of actions $\mathcal{A}$, and a time horizon $H \in \mathbb{N}$. The process starts from distribution $\mu \in \Delta(\mathcal{S})$[1], transitions via $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, emits observations via $q : \mathcal{S} \to \Delta(\mathcal{X})$, and rewards via $R : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \Delta([0,1])$. An agent-environment interaction repeatedly generates H-step trajectories $(s_1, x_1, a_1, r_1, \ldots, s_H, x_H, a_H, r_H)$ where $s_1 \sim \mu$, $s_{h+1} \sim T(\cdot|s_h, a_h)$, $x_h \sim q(s_h)$ and $r_h \sim R(x_h, a_h, x_{h+1})$ for all $h \in [H]$, and the agent chooses actions. We set $R(x_H, a_H, x_{H+1}) = R(x_H, a_H)$ for all x_H, a_H as there is no x_{H+1}. In addition, for all trajectories $\sum_{h=1}^{H} r_h \leq 1$. The agent *does not* see the states $s_1, \ldots, s_H$.

Without loss of generality, we partition $\mathcal{S}$ into subsets $\mathcal{S}_1, \ldots, \mathcal{S}_H$, where $\mathcal{S}_h$ are the only states reachable at time step h. We similarly partition $\mathcal{X}$ based on time step into $\mathcal{X}_1, \ldots, \mathcal{X}_H$. Formally, $T(\cdot \mid s, a) \in \Delta(\mathcal{S}_{h+1})$ and $q(s) \in \Delta(\mathcal{X}_h)$ when $s \in \mathcal{S}_h$. This partitioning may be internal to the agent as we can simply concatenate the time step to the states and observations. Let $\tau : \mathcal{X} \to [H]$ be the time step function, associating an observation to the time point where it is reachable.

A *policy* $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ chooses actions on the basis of observations and defines a distribution over trajectories. We use $\mathbb{E}_\pi[\cdot], \mathbb{P}_\pi[\cdot]$ to denote expectation and probability with respect to this distribution. We define the value function as:

$$\forall h \in [H], s \in \mathcal{S}_h : \; V(s;\pi) := \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s \right],$$

and policy value as $V(\pi) := \mathbb{E}_{s_1 \sim \mu}[V(s_1;\pi)]$. The goal of the agent is to find a policy that maximizes policy value. As the observation space is extremely large, we consider a function approximation setting, where the agent has access to a policy class $\Pi : \mathcal{X} \to \Delta(\mathcal{A})$. We define the class of non-stationary policies $\Pi_{\text{NS}} := \Pi^H$. A policy $\pi_{1:H} = (\pi_1, \ldots, \pi_H) \in \Pi_{\text{NS}}$ takes action a_h according to π_h.[2] The optimal policy in this class is $\pi^\star := \operatorname{argmax}_{\pi \in \Pi_{\text{NS}}} V(\pi)$, and our goal is to find a policy with value close to the optimal value, $V(\pi^\star)$.

[1] Du et al. (2019) assume the starting state is deterministic, which we generalize here.

[2] We also use h-step non-stationary policies $(\pi_1, \ldots, \pi_h) \in \Pi^h$ when we only execute this policy for h steps.

Environment assumptions. The key difference between Block MDPs and general Partially-Observed MDPs is a disjointness assumption, which removes partial observability effects and enables tractable learning.

Assumption 1. *The emission distributions for any two states $s, s' \in \mathcal{S}$ are disjoint, that is $supp(q(s)) \cap supp(q(s')) = \emptyset$ whenever $s \neq s'$.*

This disjointness assumption was argued by Du et al. (2019) to be a natural fit for visual grid-world scenarios which are common in empirical RL research. Assumption 1 allows us to define an *inverse mapping* $g^\star : \mathcal{X} \to \mathcal{S}$ such that for each $s \in \mathcal{S}$ and $x \in \text{supp}(q(s))$, we have $g^\star(x) = s$. The agent *does not* have access to $g^\star$.

Apart from disjointness, the main environment assumption is that states are reachable with reasonable probability. To formalize this, we define a *maximum visitation probability* and *reachability parameter*:

$$\eta(s) := \max_{\pi \in \Pi_{\text{NS}}} \mathbb{P}_\pi[s], \qquad \eta_{min} = \min_{s \in \mathcal{S}} \eta(s).$$

Here $\mathbb{P}_\pi[s]$ is the probability of visiting s along the trajectory taken by π. As in Du et al. (2019), our sample complexity scales polynomially with η_{min}^{-1}, so this quantity should be reasonably large. In contrast with prior work (Du et al., 2019; Dann et al., 2018), we do not require any further identifiability or determinism assumptions on the environment.

We call the policies that visit a particular state with maximum probability *homing policies*.

Definition 1 (Homing Policy)**.** *The homing policy for an observation $x \in \mathcal{X}$ is $\pi_x := \text{argmax}_{\pi \in \Pi_{NS}} \mathbb{P}_\pi[x]$. The homing policy for a state $s \in \mathcal{S}$ is $\pi_s := \text{argmax}_{\pi \in \Pi_{NS}} \mathbb{P}_\pi[s]$.*

Homing policies are *non-compositional*, in that we cannot extend homing policies for states in $\mathcal{S}_{h-1}$ to find homing policies for states in $\mathcal{S}_h$. See Appendix A for proof and further discussion. Non-compositionality implies that we must take a global policy optimization approach for learning homing policies, which we will do in the sequel.

Reward-free learning. In addition to finding a near-optimal policy, we consider a reward-free objective. In this setting, the goal is to find a small set of policies, called a *policy cover*, that we can use to visit the entire state space.

Definition 2 (Policy Cover)**.** *A finite set of non-stationary policies Ψ is called an α-policy cover if for every state $s \in \mathcal{S}$ we have $\max_{\pi \in \Psi} \mathbb{P}_\pi[s] \geq \alpha \eta(s)$.*

Intuitively, we hope to find a policy cover of size $O(|\mathcal{S}|)$. By executing each policy in turn, we can collect a dataset of observations and rewards from all states at which point it is straightforward to maximize any reward (Kakade & Langford, 2002; Munos, 2003; Bagnell et al., 2004; Antos et al., 2008; Chen & Jiang, 2019; Agarwal et al., 2019). Thus, constructing a policy cover can be viewed as an intermediate objective that facilitates reward sensitive learning.

Function classes. As the observation space is very large, we use function approximation to generalize across observations. HOMER uses two function classes. The first is the policy class $\Pi : \mathcal{X} \mapsto \Delta(\mathcal{A})$, which was used above to define the optimal value and the maximum visitation probabilities. We also use a family $\mathcal{F}_N$ of regression functions with a specific form. To define $\mathcal{F}_N$, first define $\Phi_N : \mathcal{X} \to [N]$ which maps observations into N discrete abstract states. Second, define $\mathcal{W}_N : [N] \times \mathcal{A} \times [N] \to [0, 1]$ as another "tabular" regressor class which consists of *all* functions of the specified type. Then, we set $\mathcal{F}_N := \{(x, a, x') \mapsto w(\phi^{(\text{F})}(x), a, \phi^{(\text{B})}(x')) : w \in \mathcal{W}_N, \phi^{(\text{F})}, \phi^{(\text{B})} \in \Phi_N\}$ and $\mathcal{F} := \cup_{N \in \mathbb{N}} \mathcal{F}_N$. For a simpler analysis, we assume Π and Φ_N are finite and we measure statistical complexity via $\ln|\Pi|$ and $\ln|\Phi_N|$, with no assumptions on the tabular class $\mathcal{W}_N$. Our results only involve standard uniform convergence arguments so extensions to infinite classes with other statistical complexity notions is straightforward. We emphasize that Π is typically not fully expressive.

Computational oracles. We take a "learning reductions" approach by assuming access to two well-studied learning oracles. This oracle model of computation provides no statistical benefit as the oracles can always be implemented via enumeration; the model simply serves to guide the design of practical algorithms. For the policy class Π, we assume access to an *offline contextual bandit* optimization routine:

$$\texttt{CB}(D, \Pi) := \text{argmax}_{\pi \in \Pi} \sum_{(x,a,p,r) \in D} \mathbb{E}_{a' \sim \pi(.|x)} \left[\frac{r \mathbf{1}\{a' = a\}}{p} \right].$$

The dataset consists of (x, a, p, r) quads, where $x \in \mathcal{X}$, $a \in \mathcal{A}$, $p \in [0, 1]$ and $r \in \mathbb{R}$ is the reward for the action a, which was chosen with probability p. This oracle solves a contextual bandit problem and is implementable by reduction to cost-sensitive classification (Agarwal et al., 2014).

For the regression class $\mathcal{F}_N$, we assume that we can solve *square loss minimization* problems:

$$\texttt{REG}(D, \mathcal{F}_N) := \text{argmin}_{f \in \mathcal{F}_N} \sum_{(x,a,x',y) \in D} (f(x, a, x') - y)^2.$$

Here, the dataset consists of (x, a, x', y) quads where $x, x' \in \mathcal{X}$, $a \in \mathcal{A}$ and $y \in \{0, 1\}$ is a binary label. Our function class $\mathcal{F}_N$ is non-standard due to quantization hence REG is always solving a non-convex problem. We later discuss using a standard non-quantized model class.

We assume the CB and REG oracles with n examples has a time complexity of $\text{Time}_{\text{pol}}(n)$ and $\text{Time}_{\text{reg}}(n)$ respectively.

3. Kinematic Inseparability State Abstraction

The foundational concept for our approach is a new form of state abstraction, called *kinematic inseparability*. This abstraction has three key properties demonstrated in Section 4. First, it can be learned via a reduction to supervised learning. Second, it enables reward-free exploration of the environment. Last, it enables us to learn and visualize the dynamics. We define *kinematic inseparability* below.

Definition 3 (Kinematic Inseparability)**.** *Two observations x'_1, x'_2 are* kinematically inseparable *(KI) if for every distribution $u \in \Delta(\mathcal{X} \times \mathcal{A})$ with support over $\mathcal{X} \times \mathcal{A}$ and for every $x, x'' \in \mathcal{X}$ and $a, a' \in \mathcal{A}$ the following holds:*

$$T(x'' \mid x'_1, a') = T(x'' \mid x'_2, a'), \text{ and} \tag{C1}$$
$$\mathbb{P}_u(x, a \mid x'_1) = \mathbb{P}_u(x, a \mid x'_2), \tag{C2}$$

where $\mathbb{P}_u(x, a \mid x') := \frac{T(x'|x,a)u(x,a)}{\sum_{\tilde{x},\tilde{a}} T(x'|\tilde{x},\tilde{a})u(\tilde{x},\tilde{a})}$, is the backward dynamics *measuring the probability that the previous observation and action was (x, a) given that the current observation is x' and the prior over (x, a) is u.*

Condition C1 and Condition C2 place constraints on forward dynamics (T) and backward dynamics ($\mathbb{P}_u$). We say x'_1 and x'_2 are forward KI if Condition C1 holds and backward KI if Condition C2 holds. All three notions of KI are equivalence relations, and hence they partition the observation space. The *backward kinematic inseparability dimension*, denoted $\mathrm{N_{BD}}$, is the coarsest partition size generated by the backward KI equivalence relation, with $\mathrm{N_{FD}}$ and $\mathrm{N_{KD}}$ defined similarly for the forward KI and KI relations. Partition elements represent abstract states denoted via $\phi^\star_B, \phi^\star_F, \phi^\star : \mathcal{X} \to \mathbb{N}$. For example $\phi^\star_B(x_1) = \phi^\star_B(x_2)$ if and only if x_1 and x_2 are backward KI.

For exploration, it suffices to learn backward KI. This is evident from the following lemma.

Lemma 1. *If x_1, x_2 are backward kinematic inseparable then for all $\pi_1, \pi_2 \in \Pi_{NS}$ we have $\frac{\mathbb{P}_{\pi_1}(x_1)}{\mathbb{P}_{\pi_2}(x_1)} = \frac{\mathbb{P}_{\pi_1}(x_2)}{\mathbb{P}_{\pi_2}(x_2)}$.*

The proof of this lemma and all mathematical statements in this paper are deferred to the appendices. At a high level, the lemma shows that backward KI observations induce the same ordering over policies with respect to visitation probability. This property is useful for exploration, since a policy that maximizes the probability of visiting a backward KI abstract state, also maximizes the probability of visiting each individual observation in that abstract state *simultaneously*. While backward KI is sufficient for exploration, it ignores the forward dynamics, which are useful for learning a model or visualizing the underlying dynamics.

In Appendix B, we collect and prove several useful properties of these state abstractions. We show that observations emitted from the same state are kinematically inseparable and, hence, $\max\{\mathrm{N_{FD}}, \mathrm{N_{BD}}\} \leq \mathrm{N_{KD}} \leq |\mathcal{S}|$. It is possible for $\mathrm{N_{KD}} < |\mathcal{S}|$ only when the latent state space is observationally unidentifiable. For example, if we partition the observations from a state into many "sub-states," we obtain a new Block MDP that is indistinguishable from the original. Observations from these sub-states can be shown to be kinematically inseparable. Using this, kinematic inseparability implies a canonical state space for Block MDPs.

Definition 4 (Canonical Form)**.** *A Block MDP is in* canonical form *if $\forall x_1, x_2 \in \mathcal{X}$: $g^\star(x_1) = g^\star(x_2)$ if and only if x_1 and x_2 are kinematically inseparable.*

The canonical form is simply a way to characterize the state space of a Block MDP—it does not restrict this class of environments whatsoever.

4. HOMER: Learning Kinematic Inseparability for Strategic Exploration

The main algorithm, HOMER (Algorithm 1), learns a kinematic inseparability abstraction while performing reward-free strategic exploration. Given hypothesis classes Π and $\mathcal{F}$, a positive integer N, and three hyperparameters $\eta, \epsilon, \delta \in (0, 1)$, HOMER learns a policy cover of size N and a state abstraction function for each time step. We assume $N \geq \mathrm{N_{KD}}$ and $\eta \leq \eta_{min}$ for our theoretical analysis.

HOMER operates in two phases: a reward-free phase in which it learns a policy cover (line 2-line 15) and a reward-sensitive phase where it learns a near-optimal policy for the given reward function (line 17). In the reward-free phase, HOMER proceeds inductively, learning a policy cover for time step h given the learned policy covers $\Psi_{1:h-1}$ for previous steps (line 2-line 15). In each iteration h, we first learn an abstraction function $\hat{\phi}^{(\mathrm{B})}_h$ over $\mathcal{X}_h$. This is done using a form of contrastive estimation and our function class $\mathcal{F}_N$. Specifically in the h^{th} iteration, HOMER collects a dataset D of size n_{reg} containing real and imposter transitions. We define a sampling procedure: $(x, a, x') \sim \texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$ where x is observed after rolling-in with a uniformly sampled policy in Ψ_{h-1} until time step $h-1$, action a is taken uniformly at random, and x' is sampled from $T(\cdot \mid x, a)$ (line 5). We sample two independent transitions $(x_1, a_1, x'_1), (x_2, a_2, x'_2)$ using this procedure as well as a Bernoulli random variable $y \sim \mathrm{Ber}(1/2)$. If $y = 1$ then we add the observed transition $([x_1, a_1, x'_1], y)$ to D and otherwise we add the *imposter* transition $([x_1, a_1, x'_2], y)$ (line 6-line 10). The imposter transition may not be a feasible environment outcome.

We call the subroutine REG to solve the supervised learning problem induced by D with model family $\mathcal{F}_N$ (line 11), and we obtain a predictor $\hat{f}_h = (\hat{w}_h, \hat{\phi}^{(\mathrm{F})}_{h-1}, \hat{\phi}^{(\mathrm{B})}_h)$. As we show later, $\hat{\phi}^{(\mathrm{B})}_h$ is closely related to backward KI abstraction for $\mathcal{X}_h$, and $\hat{\phi}^{(\mathrm{F})}_{h-1}$ is related to forward KI for $\mathcal{X}_{h-1}$.

Algorithm 1 HOMER$(\Pi, \mathcal{F}, N, \eta, \epsilon, \delta)$. Reinforcement and abstraction learning in a Block MDP.

1: Set $n_{\text{reg}} = \widetilde{\mathcal{O}}\left(\frac{N^6|\mathcal{A}|^3}{\eta^3}\left(N^2|\mathcal{A}| + \ln\left(\frac{|\Phi_N|H}{\delta}\right)\right)\right)$, $n_{\text{psdp}} = \widetilde{\mathcal{O}}\left(\frac{N^4H^2|\mathcal{A}|}{\eta^2}\ln\left(\frac{|\Pi|}{\delta}\right)\right)$, and $\Psi_{1:H} = \emptyset$
2: **for** $h = 2, \ldots, H$ **do**
3: $D = \emptyset$
4: **for** n_{reg} times **do**
5: $(x_1, a_1, x_1'), (x_2, a_2, x_2') \sim \texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$
6: $y \sim \text{Ber}(1/2)$
7: **if** $y = 1$ **then**
8: $D \leftarrow D \cup \{([x_1, a_1, x_1'], 1)\}$, // Real transition
9: **else**
10: $D \leftarrow D \cup \{([x_1, a_1, x_2'], 0)\}$. // Fake transition
11: $(\hat{w}_h, \hat{\phi}^{(\text{F})}_{h-1}, \hat{\phi}^{(\text{B})}_h) \leftarrow \texttt{REG}(\mathcal{F}_N, D)$ // Do Abstraction
12: **for** $i = 1$ to N **do**
13: $R_{i,h}(x, a, x') := \mathbf{1}\{\tau(x') = h \wedge \hat{\phi}^{(\text{B})}_h(x') = i\}$
14: $\pi_{i,h} \leftarrow \texttt{PSDP}(\Psi_{1:h-1}, R_{i,h}, h-1, \Pi, n_{\text{psdp}})$
15: $\Psi_h \leftarrow \Psi_h \cup \{\pi_{i,h}\}$ // Save exploration policy
16: Set $n_{\text{eval}} = \widetilde{\mathcal{O}}\left(\frac{N^2H^2|\mathcal{A}|}{\epsilon^2}\ln\left(\frac{|\Pi|}{\delta}\right)\right)$
17: $\hat{\pi} \leftarrow \texttt{PSDP}(\Psi_{1:H}, R, H, \Pi, n_{\text{eval}})$
18: **return** $\hat{\pi}$, $\Psi_{1:H}$, $\hat{\phi}^{(\text{F})}_{1:H-1}$, $\hat{\phi}^{(\text{B})}_{2:H}$

Algorithm 2 PSDP$(\Psi_{1:h}, R', h, \Pi, n)$. Optimizing reward function R' given policy covers $\Psi_{1:h}$

1: **for** $t = h, h-1, \cdots, 1$ **do**
2: $D = \emptyset$
3: **for** n times **do**
4: $(x, a, p, r) \sim \texttt{Unf}(\Psi_t) \circ \texttt{Unf}(\mathcal{A}) \circ \hat{\pi}_{t+1} \circ \cdots \circ \hat{\pi}_h$
5: $D \leftarrow \{(x, a, p, r)\} \cup D$
6: $\hat{\pi}_t \leftarrow \texttt{CB}(D, \Pi)$ // solve contextual bandit problem
7: **return** $(\hat{\pi}_1, \hat{\pi}_2, \cdots, \hat{\pi}_h)$

We define N *internal reward functions* $\{R_{i,h}\}_{i=1}^N$ corresponding to each output of $\hat{\phi}^{(\text{B})}_h$ (line 13). As argued in Section 3, backward KI is sufficient for exploration, therefore, we only use $\hat{\phi}^{(\text{B})}_h$ for defining $R_{i,h}$. The reward function $R_{i,h}$ gives a reward of 1 if the agent observes x' at time step h satisfying $\hat{\phi}^{(\text{B})}_h(x') = i$ and 0 otherwise. The internal reward functions incentivize the agent to reach different learned backward KI abstract states.

We find a policy that optimizes the internal reward functions using PSDP (Algorithm 2), which is based on Policy Search by Dynamic Programming (Bagnell et al., 2004). Using an exploratory data-collection policy, we optimize a reward function by solving a sequence of contextual bandit problems (Langford & Zhang, 2008) in a dynamic programming fashion. In our case, the policy covers for steps $1, \ldots, h-1$ induce the exploratory policy (Algorithm 2, line 4).

Formally, at time step t of PSDP, we solve

$$\max_{\pi \in \Pi} \mathbb{E}_{x_t \sim \mathcal{D}_t, a_t \sim \pi, a_{t+1:h} \sim \hat{\pi}_{t+1:h}} \left[\sum_{h'=t}^{h} R'(x_{h'}, a_{h'}, x_{h'+1})\right],$$

using the previously computed solutions $(\hat{\pi}_{t+1}, \cdots, \hat{\pi}_h)$ for future time steps. The context distribution $\mathcal{D}_t$ is obtained by uniformly sampling a policy in Ψ_t and rolling-in with it until time step t. To solve this problem, we first collect a dataset D of tuples (x, a, p, r) of size n by (1) sampling x by rolling-in with a uniformly selected policy in Ψ_t until time step t, (2) taking action a uniformly at random, (3) setting $p := 1/|\mathcal{A}|$, and (4) executing $\hat{\pi}_{t+1:h}$, and (5) setting $r := \sum_{h'=t}^{h} r_{h'}$. Then we invoke the contextual bandit oracle CB with dataset D to obtain $\hat{\pi}_t$. Repeating this process we obtain the non-stationary policy $\hat{\pi}_{1:h}$ returned by PSDP.

The learned policy cover Ψ_h for time step h is simply the policies identified by optimizing each of the N internal reward functions $\{R_{i,h}\}_{i=1}^N$. Once we find the policy covers $\Psi_{1:H}$, we perform reward-sensitive learning via a single invocation of PSDP using the *external* reward function R (Algorithm 1, line 17). In a purely reward free setting, we can just return the policy covers and learned abstractions.

We combine the two abstractions as $\overline{\phi}_h := (\hat{\phi}^{(\text{F})}_h, \hat{\phi}^{(\text{B})}_h)$ to form the learned KI abstraction, where for any $x_1, x_2 \in \mathcal{X}$, $\overline{\phi}_h(x_1) = \overline{\phi}_h(x_2)$ if and only if $\hat{\phi}^{(\text{F})}_h(x_1) = \hat{\phi}^{(\text{F})}_h(x_2)$ and $\hat{\phi}^{(\text{B})}_h(x_1) = \hat{\phi}^{(\text{B})}_h(x_2)$. We define $\hat{\phi}^{(\text{B})}_1(x) \equiv 1$ and $\hat{\phi}^{(\text{F})}_H \equiv 1$ as there is no backward and forward dynamics at these steps, respectively. Empirically, we use $\overline{\phi}$ for learning the transition dynamics and visualization (see Section 7).

5. Theoretical Analysis

Our main theoretical contribution is to show that HOMER computes a policy cover and a near-optimal policy with high probability in a sample-efficient and computationally-tractable manner. The result requires an additional expressivity assumption on classes Π and $\mathcal{F}$, which we now state.

Assumption 2. *Let* $\mathcal{R} := \{R\} \cup \{(x, a, x') \mapsto \mathbf{1}\{\phi(x') = i \wedge \tau(x') = h\} \mid \phi \in \Phi_N, i \in [N], h \in [H], N \in \mathbb{N}\}$ *be the set of external and internal reward functions. We assume that* Π *satisfies* policy completeness *for every* $R' \in \mathcal{R}$*: for every* $h \in [H]$ *and* $\pi' \in \Pi_{NS}$*, there exists* $\pi \in \Pi$ *such that for each* $x \in \mathcal{X}_h$ *we have:*

$$\pi(x) = \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid x_h = x, a_h = a, a_{h'} \sim \pi'\right].$$

We also assume that $\mathcal{F}$ *is* realizable*: for any* $h \in [H]$*,* $N \geq N_{KD}$*, and any prior distribution* $\rho \in \Delta(\mathcal{S}_h)$ *with*

supp$(\rho) = \mathcal{S}_h$*, there exists* $f_\rho \in \mathcal{F}_N$*, such that for any* $x \in \mathcal{X}_{h-1}, a \in \mathcal{A}$*, and* $x' \in \mathcal{X}_h$ *we have:*

$$f_\rho(x, a, x') = \frac{T(g^\star(x')|g^\star(x), a)}{T(g^\star(x')|g^\star(x), a) + \rho(g^\star(x'))}.$$

Completeness assumptions are common in the analysis of dynamic programming style algorithms for the function approximation setting (Antos et al., 2008) (see Chen & Jiang (2019) for a detailed discussion). Our exact completeness assumption appears in the work of Dann et al. (2018), who use it to derive an efficient algorithm for a restricted version of our setting with deterministic latent state transitions.

The realizability assumption on $\mathcal{F}$ is adapted to our learning approach: as we use $\mathcal{F}$ to distinguish between real and imposter transitions, $\mathcal{F}$ should contain the optimal regressor for these problems. In HOMER, the sampling procedure we use to collect data for the learning problem in the h^{th} iteration induces a marginal distribution $\rho \in \Delta(\mathcal{S}_h)$ and the optimal regressor for this problem is f_ρ (See Lemma 9 in Appendix D). It is not hard to see that if x_1, x_2 are kinematically inseparable then $f_\rho(x_1, a, x') = f_\rho(x_2, a, x')$ and the same claim holds for the third argument of f_ρ. Therefore the realizability structure of $\mathcal{F}_N$ ensures that Φ_N contains a kinematic inseparability abstraction.

Theoretical Guarantees. We now state the main guarantee.

Theorem 1 (Main Result)**.** *For any Block MDP and hyperparameters* $\epsilon, \delta, \eta \in (0, 1), N \in \mathbb{N}$*, satisfying* $\eta \leq \eta_{min}$ *and* $N \geq N_{KD}$*,* HOMER *outputs exploration policies* $\Psi_{1:H}$ *and a reward sensitive policy* $\hat{\pi}$ *satisfying:*

1. Ψ_h *is an* $1/2$*-policy cover of* $\mathcal{S}_h$ *for every* $h \in [H]$
2. $V(\hat{\pi}) \geq \max_{\pi \in \Pi_{NS}} V(\pi) - \epsilon$

with probability least $1 - \delta$*. The sample complexity of* HOMER *is* $\mathcal{O}\left(n_{\text{psdp}} N H^3 + n_{\text{reg}} H + n_{\text{eval}} H\right)$ *where* $n_{\text{psdp}}, n_{\text{reg}}, n_{\text{eval}}$ *are as specified in Algorithm 1, which gives*

$$\widetilde{\mathcal{O}}\left(\frac{N^8|\mathcal{A}|^4 H}{\eta^3} + \frac{N^6|\mathcal{A}|H}{\eta^3}\ln(|\Phi_N|/\delta) + \left(\frac{N^5 H^4|\mathcal{A}|}{\eta^2} + \frac{N^2 H^3|\mathcal{A}|}{\epsilon^2}\right)\ln(|\Pi|/\delta)\right).$$

The running time is $\mathcal{O}\left(n_{\text{psdp}} N H^3 + n_{\text{reg}} H^2 + n_{\text{eval}} H^2 + \text{Time}_{\text{pol}}(n_{\text{psdp}}) N H^2 + \text{Time}_{\text{reg}}(n_{\text{reg}}) H + \text{Time}_{\text{pol}}(n_{\text{eval}}) H\right)$.

Theorem 1 shows that executing HOMER with $\text{N}_{\text{KD}} \leq N \leq c\text{N}_{\text{KD}}$ and $\frac{\eta_{min}}{d} \leq \eta \leq \eta_{min}$ for some constants $c, d \geq 1$, gives us a sample complexity of $\text{poly}(\text{N}_{\text{KD}}, H, |\mathcal{A}|, \eta_{min}^{-1}, \epsilon^{-1}, \log|\Pi|/\delta)$, which at a coarse level is our desired scaling. Empirically, we can set the hyperparameters by running HOMER with $N = 2^t$ and $\eta = \frac{1}{2^t}$ for increasing values of t, and stopping when the final learned policy stops improving. Recall that $\text{N}_{\text{KD}} \leq |\mathcal{S}|$, hence our bounds are polynomially dependent on the state space but crucially do not depend upon the size of observation space. Further, our bounds only depend on $\log|\Phi_N|$ which means we can use an exponentially large model family for Φ_N. In terms of computation, the running time is polynomial in our oracle model, where we assume we can solve contextual bandit problems over Π and regression problems over $\mathcal{F}_N$. In Section 7, we see that these problems can be solved effectively in practice.

The closest related result is for the PCID algorithm of Du et al. (2019). PCID provide guarantees only for a restricted class of Block MDPs. The precise details of the guarantee differs from ours in several ways (e.g., additive versus multiplicative error in policy cover definition, different computational and expressivity assumptions), so the sample complexity bounds are incomparable. However, Theorem 1 represents a significant conceptual advance as it eliminates the identifiability assumptions required by PCID and therefore greatly increases the scope for tractable RL.

Why does HOMER learn kinematic inseparability? A detailed proof of Theorem 1 is deferred to Appendix C-Appendix D, but for intuition, we provide a sketch of how HOMER learns a kinematic inseparability abstraction. For this discussion only, we focus on asymptotic behavior and ignore sampling issues.

Inductively, assume that Ψ_{h-1} is a policy cover of $\mathcal{S}_{h-1}$, let $D(x, a, x')$ be the marginal distribution over real and imposter transitions sampled by HOMER in the h^{th} iteration (line 4–line 10), and let ρ be the marginal distribution over $\mathcal{X}_h$. First observe that as Ψ_{h-1} is a policy cover, we have $\text{supp}(D) = \mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h$, which is crucial for our analysis. Let $\hat{f} = (\hat{w}_h, \hat{\phi}_{h-1}^{(\text{F})}, \hat{\phi}_h^{(\text{B})})$ be the output of the regression oracle REG in this iteration. The first observation is that the Bayes optimal regressor for this problem is f_ρ defined in Assumption 2, and, with realizability, in this asymptotic discussion we have $\hat{f} \equiv f_\rho$.

Next, we show that for any two observations $x_1', x_2' \in \mathcal{X}_h$, if $\hat{\phi}_h^{(\text{B})}(x_1') = \hat{\phi}_h^{(\text{B})}(x_2')$ then x_1' and x_2' are backward kinematically inseparable. If this precondition holds, then $\forall x \in \mathcal{X}_{h-1}, a \in \mathcal{A}$ we have:

$$f_\rho(x, a, x_1') = \hat{f}(x, a, x_1') = \hat{w}_h(\hat{\phi}_{h-1}^{(\text{F})}(x), a, \hat{\phi}_h^{(\text{B})}(x_1')) = \hat{w}_h(\hat{\phi}_{h-1}^{(\text{F})}(x), a, \hat{\phi}_h^{(\text{B})}(x_2')) = \hat{f}(x, a, x_2') = f_\rho(x, a, x_2').$$

Then, by inspection of the form of f_ρ, we have

$$f_\rho(x, a, x_1') = f_\rho(x, a, x_2') \Leftrightarrow \frac{T(x_1' \mid x, a)}{\rho(x_1')} = \frac{T(x_2' \mid x, a)}{\rho(x_2')}.$$

As this identity holds for all $x \in \mathcal{X}_{h-1}, a \in \mathcal{A}$ and trivially when $x \notin \mathcal{X}_{h-1}$, it is easy to see that x_1', x_2' are backward

KI. Formally, for any prior $u \in \Delta(\mathcal{X}, \mathcal{A})$, we have

$$\mathbb{P}_u(x, a \mid x'_1) = \frac{T(x'_1 \mid x, a)u(x, a)}{\sum_{\tilde{x},\tilde{a}} T(x'_1 \mid \tilde{x}, \tilde{a})u(\tilde{x}, \tilde{a})}$$
$$= \frac{\frac{\rho(x'_1)}{\rho(x'_2)} T(x'_2 \mid x, a)u(x, a)}{\sum_{\tilde{x},\tilde{a}} \frac{\rho(x'_1)}{\rho(x'_2)} T(x'_2 \mid \tilde{x}, \tilde{a})u(\tilde{x}, \tilde{a})} = \mathbb{P}_u(x, a \mid x'_2).$$

This implies that $\hat{\phi}_h^{(\mathrm{B})}$ is a backward KI abstraction over $\mathcal{X}_h$. Similarly, we can show that $\hat{\phi}_{h-1}^{(\mathrm{F})}$ is a forward KI abstraction over $\mathcal{X}_{h-1}$ (See Appendix D.4 for proof).

Standardizing `REG` Oracle. We learn abstractions by solving regression problems with the quantized model class $\mathcal{F}_N$. While this is empirically feasible as we will see, it always result in a difficult optimization problem and requires a particular form for the model class. We show how to avoid this in Appendix E, where we present a parallel version of our algorithm and guarantees using a black-box (non-quantized) regression class. The main algorithmic difference is that we recover the abstraction by clustering the outputs of the predictor trained to distinguish real and imposter transitions.

Limitation of Existing Abstractions. In Appendix G we present examples showing that strategies for learning abstraction from prior work can lead to exploration failures. We specifically demonstrate failures for (a) predicting the previous action (Pathak et al., 2017), (b) predicting the previous abstract state and action (Du et al., 2019), and (c) using autoencoders (Tang et al., 2017). Figure 2a provides a sketch of the autoencoding failure. If observations contain a bit encoding the state along with many more noisy bits, the optimal autoencoder will memorize a noise bit and ignore the state. This naturally leads to exploration failure.

6. Related Work

Sample efficient exploration of Markov Decision Processes with a small number of observed states has been studied in a number of papers (Brafman & Tennenholtz, 2002; Strehl et al., 2006; Jaksch et al., 2010), initiated by the breakthrough result of Kearns & Singh (2002). While state-of-the-art results provide near-optimal guarantees for these small-state MDPs, the algorithms do not exploit latent structures, and therefore, cannot scale to the rich-observation environments that are popular in modern empirical RL.

A recent line of theoretical work (Krishnamurthy et al., 2016; Jiang et al., 2017) focusing on rich observation reinforcement learning has shown that it is information-theoretically possible to explore these environments and has provided computationally efficient algorithms for some special settings. In particular, Dann et al. (2018) considers deterministic latent-state dynamics while Du et al. (2019) allows for limited stochasticity. As we have mentioned, the present work continues in this line by eliminating assumptions required by these results, further expanding the scope for tractable rich observation reinforcement learning. Specifically, compared to the `PCID` algorithm of Du et al. (2019), `HOMER` can handle a stochastic start state and does not require any margin assumptions on the Block MDP. In addition, our algorithm does not rely on abstract states for defining policies or future prediction problems which avoids cascading errors due to inaccurate predictions.

On the empirical side, a number of approaches have been proposed for exploration with large observation spaces using pseudo-counts (Tang et al., 2017), optimism-driven exploration (Chen et al., 2017), intrinsic motivation (Bellemare et al., 2016), and prediction errors (Pathak et al., 2017). While these algorithms can perform well on certain RL benchmarks, we lack a deep understanding of their behavior and failure modes. As the earlier discussion and examples in Appendix G show, using the representations learned by these methods for provably efficient exploration is challenging, and may not be possible in some cases.

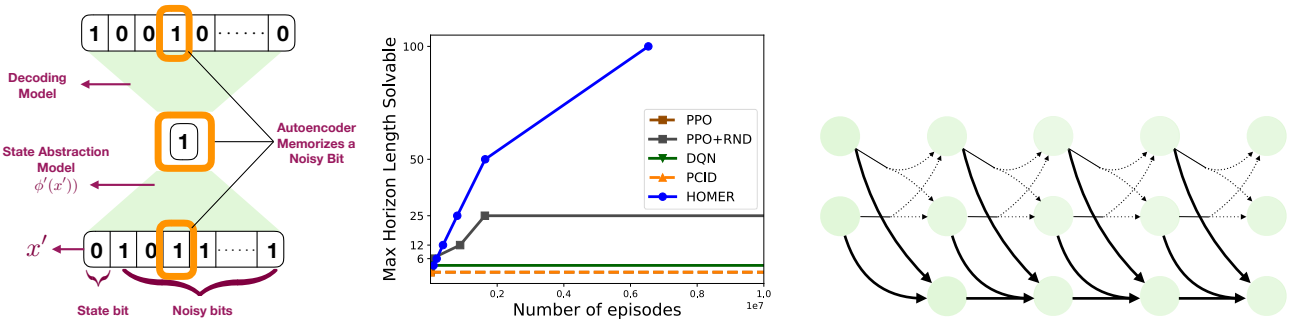
Most closely related to our work, Nachum et al. (2019) use a supervised learning objective similar to ours for learning state abstractions. However, they do not address the problem of exploration and do not provide any sample complexity guarantees. Importantly, we arrive at our supervised learning objective with the goal to learn kinematic inseparability.

7. Proof of Concept Experiments

We evaluate on a challenging problem called a *diabolical combination lock* that contains high-dimensional observations, precarious dynamics, and anti-shaped, sparse rewards.

The environment. The diabolical combination lock is a class of rich observation MDPs. For a fixed horizon H and action space size K, the state space is given by $\mathcal{S} := \{s_{1,a}, s_{1,b}\} \cup \{s_{h,a}, s_{h,b}, s_{h,c}\}_{h=2}^{H}$ and the action space by $\mathcal{A} := \{a_1, ..., a_K\}$. The agent starts in either $s_{1,a}$ or $s_{1,b}$ with equal probability. After taking h actions the agent is in $s_{h+1,a}, s_{h+1,b}$ or $s_{h+1,c}$. Informally, the states $\{s_{h,a}\}_{h=1}^{H}$ and $\{s_{h,b}\}_{h=1}^{H}$ are "good" states from which optimal return is achievable, while the states $\{s_{h,c}\}_{h=2}^{H}$ are "bad" states from which an optimal return is impossible. Each good state has a single good action, denoted u_h for $s_{h,a}$ and v_h for $s_{h,b}$, which transitions the agent uniformly to one of the two good states at the next time step. All other good state actions and all bad state actions lead to the bad state at the next time. We fix the vectors $u_{1:H}, v_{1:H}$ before the learning process by choosing each action uniformly from $\mathcal{A}$.

The agent receives a reward of 5 on taking action u_H in $s_{H,a}$ or action v_H in $s_{H,b}$. Upon transitioning from one good state to another good state at time step $h \in [H-1]$, the agent receives an anti-shaped reward of $-1/(H-1)$. For



(a) (b) (c)

Figure 2: **Left:** Failure case for autoencoder training (see text and Appendix G for full discussion). **Center:** Results on the diabolical combination lock problem showing horizon against number of episodes needed to achieve mean return of $V(\pi^\star)/2$. **Right:** Dynamics and abstraction for first 4 steps, learned by HOMER for $H = 100$ and $K = 10$.

many algorithms this structure leads the agent away from the optimal policy. The agent receives a reward of 0 for all other transitions. We have $\eta_{min} = 1/2$ and $V(\pi^\star) = 4$.

The agent never directly observes the state and instead receives an observation $x \in \mathbb{R}^d$ where $d = 2^{\lceil \log_2(H+4) \rceil}$, generated stochastically. We add mean 0 and variance 0.1 Gaussian noise to a 2-sparse vector encoding the state and timestep identity, then multiply with a Hadamard matrix. See Appendix H for full details and environment figure.

Our main experiments consider $H = 100$ and $|\mathcal{A}| = K = 10$. In this case, the problem is googol-sparse: the probability of finding the optimal return through random search is 10^{-100}.[3] Moreover, for any fixed sequence of actions the probability of an optimal return is at most $2^{-\tau}$ where $\tau := \sum_{h=1}^{100} \mathbf{1}\{u_h \neq v_h\}$. As $u_{1:H}$ and $v_{1:H}$ are chosen randomly, we have $\mathbb{E}[\tau] = 90$ in these instances.

HOMER implementation. We use non-stationary deterministic policies, where each policy is represented as a tuple of H linear models $\pi = (W_1, W_2, \cdots, W_H)$. Here $W_h \in \mathbb{R}^{|\mathcal{A}| \times d}$ for each $h \in [H]$. Given an observation $x \in \mathbb{R}^d$ at time step h, the policy takes the action $\pi(x) := \text{argmax}_{a \in \mathcal{A}} (W_h x)_a$. We represent a state abstraction function $\phi : \mathcal{X} \to [N]$ using a linear model $B \in \mathbb{R}^{N \times d}$. Given an observation x we decode it to the abstract state $\phi(x) = \arg\max_{i \in [N]} (Bx)_i$. The regressor class $\mathcal{F}$ uses a two-layer neural network with ReLu non-linearity and a Gumbel Softmax operation on the output of $\phi(x)$ to make the model end-to-end differentiable. We make a few implementation changes for empirical efficiency of HOMER without changing key ideas. We provide full details of the model, optimization and empirical changes in Appendix H.

Baselines. We compare our method against Proximal Policy Optimization (PPO) (Schulman et al., 2017). PPO uses a naive exploration strategy based on entropy bonus which is often insufficient for challenging exploration problems. Therefore, we also augment it with an exploration bonus based on Random Network Distillation (RND) (Burda et al., 2019), denoted PPO + RND. We also compare against Deep Q Networks (DQN) (Mnih et al., 2015) which are a value function method. Lastly, we consider the model-based algorithm (PCID) of Du et al. (2019). Their approach makes certain margin assumptions on the MDP which are violated by this problem. We use publicly available code for running these baselines. For details see Appendix H.

Results. Figure 2b reports the minimum number of episodes needed to achieve a mean return of $V(\pi^\star)/2 = 2.0$. We run each algorithm 5 times with different seeds and for a maximum of 10 million episodes, and we report the median performance. We run each method on increasingly longer horizons until it fails to achieve a mean return of 2. As we can see, PPO fails at $H = 3$ and DQN at $H = 6$ as expected given their simple exploration methods. Adding RND bonus is helpful, and PPO + RND can solve problems with $H = 25$, but it fails at $H = 50$. PCID fails at $H = 3$ showing that its margin assumption is empirically limiting. Finally, HOMER is able to solve the problem for all horizons. Figure 2c shows the recovered dynamics for the first four steps. The top two rows show the "good states" and the bottom row shows the "bad states." HOMER is able to accurately find the canonical form of the Block MDP, and using count-based statistics we estimate the transition probabilities up to a maximum error of 0.03. In Appendix H, we show the error bars, and visualize the visitation probabilities.

We plot the moving average of returns against the number of episodes on the diabolical combination lock problem with $H = 100$ and $K = 10$ in Figure 3. We compare the performance of HOMER against the best baseline PPO + RND. The result shows that HOMER is able to learn the optimal policy while PPO + RND fails to do so. Furthermore, the plot of HOMER shows three distinct regions. The first region up

[3] For comparison, 10^{100} is more than the current estimate of the total number of elementary particles in the universe.

Statistics	$N = 1$	$N = 2$	$N = 3$	$N = 4$
Max	∞	6.55×10^6	6.65×10^6	6.71×10^6
Median	∞	6.54×10^6	6.65×10^6	6.7×10^6
Min	∞	6.53×10^6	6.63×10^6	6.69×10^6

Table 1: Performance of HOMER on diabolical combination lock with $H = 100$ and $K = 10$. We vary the abstract state space size (N) and report the number of episodes needed to achieve a mean return of $V(\pi^\star)/2$. We report median, max and min performance over five runs with different seeds. If the algorithm fails to solve the problem in 10^7 episodes then we report the result as ∞ indicating timeout.

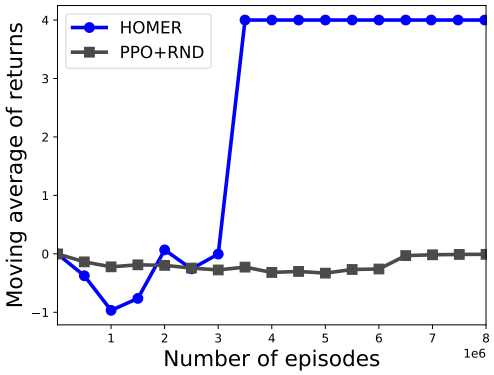


Figure 3: Results on diabolical combination lock with horizon (H) of 100 and action space (K) of size 10. We plot the moving average of returns against the number of episodes for HOMER and PPO + RND. We have $V(\pi^\star) = 4.0$

to 10^6 episodes shows a decline in return as the algorithm learns to explore. This is due to the negative antishaped reward which occurs when moving from one good state to the next. The second region between 10^6 and 3×10^6 episodes is when the algorithm is learning a reward-sensitive policy. This region shows an increase in returns. The last region is when the algorithm is exploiting using the learned policies and this consistently gives an optimal return of 4.

Performance on varying abstract state space size (N). HOMER uses two hyperparameters: the size of the abstract state space N and an estimate η of the reachability parameter η_{min}. In our main experiments, we implicitly search over η by using different values of n_{reg} and n_{psdp}, but we always use $N = 2$. We study the performance when varying N by running HOMER five times on different seeds for different values of N. We set the other hyperparameters to the best setting for $H = 100$ and $K = 10$. Results are given in Table 1. We fail to solve the problem with $N = 1$, which is expected since the entire observation space is mapped to the same abstract state. However, we consistently solve the problem for $N \geq 2$. This is consistent with our theoretical results where the only constraint on N is that it should be greater than N_{KD}. The diabolical combination lock has two backward KI abstract states at each timestep: one corresponding to the two good states $\{s_{h,a}, s_{h,b}\}$ and the other corresponding to the bad state $s_{h,c}$. Hence, $N \geq 2$ is sufficient on a per timestep basis. Furthermore, we see that HOMER does not use significantly more episodes when doubling N from 2 to 4.

Reproducibility. Code and models can be found at https://github.com/cereb-rl.

8. Conclusion

We present HOMER, a model-free RL algorithm for rich observation environments. We prove theoretical guarantees for HOMER and provide proof of concept experiments on a challenging domain. Applying HOMER to real-world RL scenarios is a future work direction.

Acknowledgements. We thank Miro Dudik for suggesting the oracle without bottleneck structures in Appendix E. We thank Qinghua Liu for helpful feedback on the proof. We thank Miro Dudik, Alekh Agarwal, Wen Sun, and Nan Jiang for useful discussion. We thank Vanessa Milan for help with Figures. We also thank Microsoft Philly Team for providing us with computational resources and help for running experiments.

References

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. E. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 2014.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in markov decision processes. *arXiv:1908.00261*, 2019.

Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 1997.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimiza-

tion based fitted policy iteration and a single sample path. *Machine Learning*, 2008.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, 2017.

Bagnell, J. A., Kakade, S. M., Schneider, J. G., and Ng, A. Y. Policy search by dynamic programming. In *Advances in Neural Information Processing Systems*, 2004.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

Brafman, R. I. and Tennenholtz, M. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 2002.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019.

Chen, R. Y., Sidor, S., Abbeel, P., and Schulman, J. UCB exploration via Q-Ensembles. *arXiv:1706.01502*, 2017.

Dann, C., Lattimore, T., and Brunskill, E. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Dann, C., Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. On oracle-efficient PAC RL with rich observations. In *Advances in Neural Information Processing Systems*, 2018.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. https://github.com/openai/baselines, 2017.

Du, S. S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudík, M., and Langford, J. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019.

Givan, R., Dean, T., and Greig, M. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 2003.

Hazan, E., Kakade, S. M., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. *arXiv:1812.02690*, 2018.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 2010.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2016.

Jiang, N. Notes on state abstractions. http://nanjiang.cs.illinois.edu/files/cs598/note4.pdf, 2018.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.

Jong, N. K. and Stone, P. State abstraction discovery from irrelevant state variables. In *International Joint Conference on Artificial Intelligence*, 2005.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, University College London, 2003.

Kakade, S. M. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 2002.

Krishnamurthy, A., Agarwal, A., and Langford, J. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, 2016.

Langford, J. and Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, 2008.

Lattimore, T. and Hutter, M. PAC bounds for discounted MDPs. In *Conference on Algorithmic Learning Theory*, 2012.

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

Liang, T., Rakhlin, A., and Sridharan, K. Learning with square loss: Localization through offset Rademacher complexity. In *Conference on Learning Theory*, 2015.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

Mccallum, A. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, The University of Rochester, 1996.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 2015.

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

Modi, A., Jiang, N., Tewari, A., and Singh, S. Sample complexity of reinforcement learning using linearly combined model ensembles. *arXiv:1910.10597*, 2019.

Munos, R. Error bounds for approximate policy iteration. In *International Conference on Machine Learning*, 2003.

Nachum, O., Gu, S., Lee, H., and Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2019.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017.

Ravindran, B. *An Algebraic Approach to Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2004.

Ross, S. and Bagnell, J. A. Reinforcement and imitation learning via interactive no-regret learning. *arXiv:1406.5979*, 2014.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Shangtong, Z. Modularized implementation of deep RL algorithms in PyTorch. https://github.com/ShangtongZhang/DeepRL, 2018.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 2008.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, 2006.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Tieleman, T. and Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning, 2012.