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.

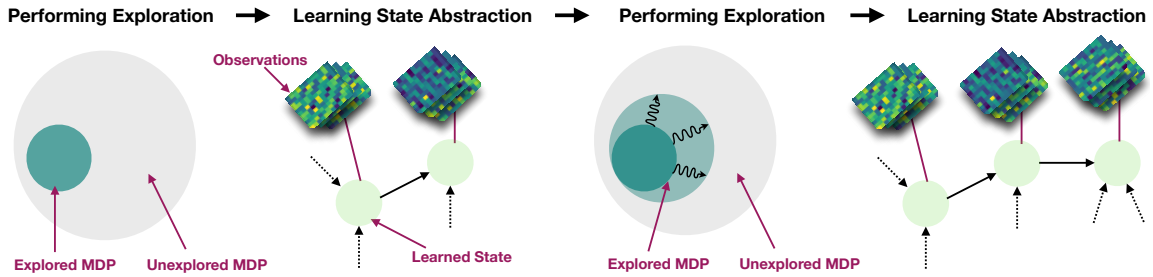


Figure 1: HOMER learns a set of exploration policies and a state abstraction function by iterating between exploring using the current state abstraction and refining the state abstraction based on the new experience.

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 $\Psi_{1:h}$

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

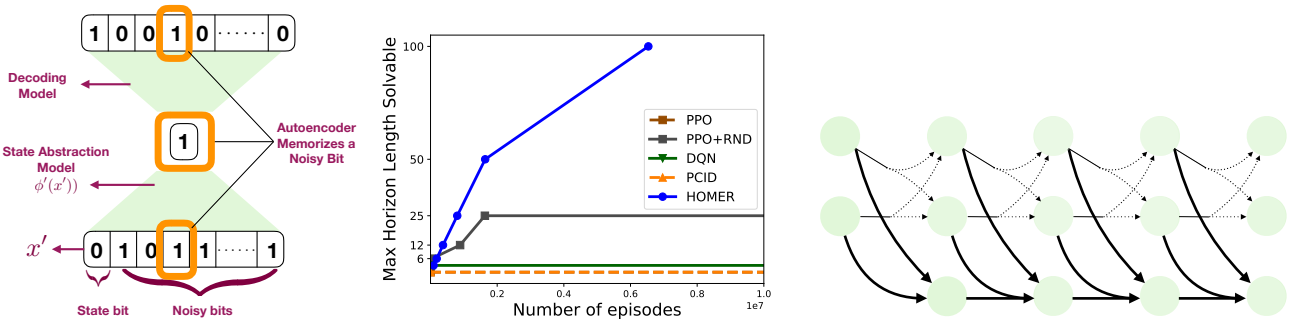
Most closely related to our work, Nachum et al. (2019) use a supervised learning objective similar to ours for learning state abstractions. However, they do not address the problem of exploration and do not provide any sample complexity guarantees. Importantly, we arrive at our supervised learning objective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

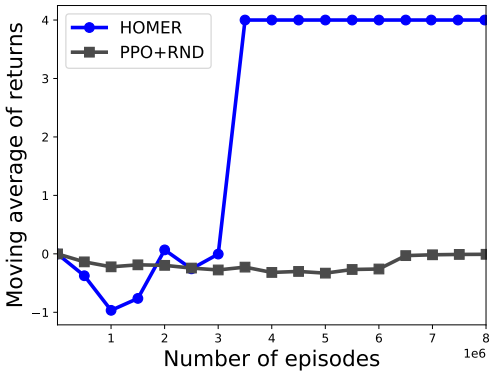


Figure 3: Results on diabolical combination lock with horizon (H) of 100 and action space (K) of size 10. We plot the moving average of returns against the number of episodes for HOMER and PPO + RND. We have $V(\pi^\star) = 4.0$

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.

Appendices

Symbol	Definition
$[N]$	Defines the set $\{1, 2, \cdots, N\}$ for any $N \in \mathbb{N}$.
$\Delta(\mathcal{U})$	The space of probability distribution over the set $\mathcal{U}$.
$\texttt{Unf}(\mathcal{U})$	A uniform distribution over the set $\mathcal{U}$.
$\text{supp}(p)$	Support of a distribution. For any $p \in \Delta(\mathcal{U})$, we have $\text{supp}(p) := \{u \in \mathcal{U} \mid p(u) > 0\}$.
$\mathcal{M}$	Denotes a Block Markov Decision Process (Block MDP).
H	Number of actions the agent takes for any episode.
$\mathcal{S}$	A finite state space of $\mathcal{M}$. The process is layered, so states also encode the time step.
$\mathcal{S}_h$	The set of states reachable at time step h. $\mathcal{S} := \cup_{h\in[H]}\mathcal{S}_h$.
$\mathcal{X}$	The observation space. May be infinitely large, but is assumed to be countable.
$\mathcal{X}_h$	Set of observations reachable at time step h. $\mathcal{X} := \cup_{h\in[H]}\mathcal{X}_h$.
$\mathcal{A}$	The finite discrete action space of $\mathcal{M}$
q	An emission function $q : s \to \Delta(\mathcal{X})$. $q(x \mid s)$ denotes the probability of observing x in state s. Note that $\text{supp}(q(\cdot\|s)) \cap \text{supp}(q(\cdot\|s')) = \emptyset$ when $s \neq s'$.
$g^\star$	A decoder function $g^\star : \mathcal{X} \to \mathcal{S}$. $g^\star(x) = s$ iff $q(x\|s) > 0$.
$\mu(s)$	Probability of starting in state s at the beginning of any episode.
$T(s' \mid s, a)$	The probability of transitioning to $s' \in \mathcal{S}$ when taking action $a \in \mathcal{A}$ in $s \in \mathcal{S}$.
π	A policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$, which may or may not be stationary.
$(\pi_1, \ldots, \pi_h)$	A h-step policy where the t^{th} action ($1 \le t \le h$) is taken according to π_t.
Π	The policy class, $\Pi : \mathcal{X} \to \Delta(\mathcal{A})$.
Π_{NS}	The set of non-stationary policies: $\Pi_{\text{NS}} := \{(\pi_1, \ldots, \pi_H) : \pi_t \in \Pi\}$.
$\mathbb{P}_\pi(s)$	Probability of visiting s when following π, from the starting distribution μ.
$\pi^\star_s$	Homing policy of the state s, $\pi^\star_s := \arg\max_{\pi\in\Pi_{\text{NS}}} \mathbb{P}_\pi(s)$. Due to Lemma 2, we take $\pi^\star_s$ to be deterministic and non-stationary.
$\pi^\star_x$	Analogous homing policy of the observation x, $\pi^\star_x := \arg\max_{\pi\in\Pi_{\text{NS}}} \mathbb{P}_\pi(x)$ It is easy to see from the Block MDP assumption that $\pi^\star_x = \pi^\star_s$ where $s = g^\star(x)$.
$V(\pi; R)$	Value for (non-stationary) policy π on reward function R from starting distribution μ. R may have type $R : \mathcal{X} \to \mathbb{R}$ or $R : (\mathcal{X} \times \mathcal{A}) \to \mathbb{R}$ and may also be stochastic.
$V(s; \pi, R)$ and $V(x; \pi, R)$	Value functions for π on R, defined over states or observations.
$\eta(s)$	Maximum visitation probability for state s, $\eta(s) := \sup_{\pi\in\Pi} \mathbb{P}_\pi(s)$.
η_{min}	Reachability parameter, $\eta_{min} := \min_{s\in\mathcal{S}} \eta(s)$.
$\mathcal{F}$	Regressor class containing functions f from $\mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0, 1]$.
Φ_N	Decoder function class $\Phi_N : \mathcal{X} \to [N]$, for a fixed N.
τ	Time function that maps states/observations to the time step where they are reachable, which is well-defined due to the layered assumption.

Table 2: List of notations and brief definitions. Operators defined on states are lifted to observations in the natural way, e.g., $\mu(x) := \sum_s \mu(s)q(x \mid s)$.

Notation and Overview. See Table 2 for an overview of the notation and definitions used in this appendix. The appendix is structured as follows:

- Appendix A: Properties of homing policies;
- Appendix B: Properties of kinematic inseparability;
- Appendix C: Basic results for Policy Search from Dynamic Programming (PSDP);
- Appendix D: Analysis of the HOMER algorithm;
- Appendix E: Recovering state abstraction from non-quantized model class via clustering;
- Appendix F: Supporting results;
- Appendix G: Failure Examples for Existing Methods;
- Appendix H: Experimental setup, optimization details and additional results.

A. Properties of Homing Policies

In this section we prove some basic properties of homing policies. For this section only, we consider a fully expressive policy set $\Upsilon := (\mathcal{X} \to \Delta(\mathcal{A}))$. We further define the set of *all* deterministic policies $\Upsilon_{\text{det}} := (\mathcal{X} \to \mathcal{A})$. Clearly we have $\Upsilon_{\text{det}} \subset \Upsilon$. Note that both of these classes contain non-stationary policies due to the layered structure of the environments we consider. In particular, we have $(\pi_1, \ldots, \pi_H) \in \Upsilon$ whenever $\pi_h \in \Upsilon$ for all h, with a similar statement for Υ_{det}.

The first result is that for every state, there exists a *deterministic non-stationary* policy $\pi \in \Upsilon_{\text{det}}$ that is a homing policy for that state. This motivates our decision to restrict our search to only these policies in experiments. The result also appears in (Bagnell et al., 2004), but we provide a proof for completeness.

Lemma 2. *For any, possibly stochastic, reward function R, we have*

$$\max_{\pi \in \Upsilon_{det}} V(\pi; R) = \max_{\pi \in \Upsilon} V(\pi; R),$$

where $V(\pi; R)$ is the value for policy π under reward function R. In particular, the result holds for $R(x) = \mathbf{1}\{g^\star(x) = s\}$, for any $s \in \mathcal{S}$, which yields:

$$\max_{\pi \in \Upsilon_{det}} \mathbb{P}_\pi(s) = \max_{\pi \in \Upsilon} \mathbb{P}_\pi(s).$$

Proof. As $\Upsilon_{\text{det}} \subset \Upsilon$, that the LHS is at most the RHS is obvious. We are left to establish the other direction.

The proof is a simple application of dynamic programming. Assume inductively that there exists a policy $\tilde{\pi}_{h:H} \in \Upsilon_{\text{det}}$ such that, for any distribution $Q \in \Delta(\mathcal{X}_h)$, we have

$$\mathbb{E}_{x_h \sim Q}\left[V(\tilde{\pi}_{h:H}; R, x_h)\right] \geq \max_{\pi_{h:H} \in \Upsilon} \mathbb{E}_{x_h \sim Q}\left[V(\pi_{h:H}; R, x_h)\right],$$

where the value function here denotes the future reward, according to R, when executing the policy from the starting observation x_h. The base case is when $h = H$, in which case, it is easy to verify that the claim holds for the policy $\tilde{\pi}_H(x_H) := \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}[R \mid x_H]$.

Define the policy component for time step $h-1$ as

$$\forall x_{h-1} \in \mathcal{X}_{h-1} : \tilde{\pi}_{h-1}(x_{h-1}) := \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}[R + V(\tilde{\pi}_{h:H}; R, x_h) \mid x_{h-1}].$$

Now for any potentially stochastic policy $\pi_{h-1:H}$, and any distribution $Q \in \Delta(\mathcal{X}_{h-1})$ we have

$$\begin{aligned}
\mathbb{E}_{x_{h-1} \sim Q}\left[V(\tilde{\pi}_{h-1:H}; R, x_{h-1})\right] &\geq \mathbb{E}_{x_{h-1} \sim Q}\left[R + V(\tilde{\pi}_{h:H}; R, x_h) \mid a \sim \pi_{h-1}\right] \\
&\geq \mathbb{E}_{x_{h-1} \sim Q}\left[R + V(\pi_{h:H}; R, x_h) \mid a \sim \pi_{h-1}\right] \\
&= \mathbb{E}_{x_{h-1} \sim Q}\left[V(\pi_{h:H}; R, x_{h-1})\right].
\end{aligned}$$

This proves the inductive step. We conclude the proof by noting that $V(\pi; R) = \mathbb{E}_{x_1 \sim \mu} V(\pi; R, x_1)$, for which we have established the optimality guarantee for $\tilde{\pi}_{1:H}$. □

We also observe that homing policies do not grow compositionally. In other words, we may not be able to construct homing policies for states $\mathcal{S}_h$, by appending a one-step policy to the homing policies for $\mathcal{S}_{h-1}$. Note that this holds even when working with the unrestricted policy class Υ. This observation justifies the global policy search procedure PSDP for finding the homing policies.

For the statement, for a policy subset Π', we use the notation $\Delta(\Pi')$ to denote the set of *mixture policies* that, on each episode samples a policy $\pi \in \Pi'$ from a distribution and executes that policy. Note that this is not the same as choosing a new policy from the distribution on a per time-step basis.

Lemma 3. *There exists a Block MDP $\mathcal{M}$ a time step $h \in [H]$ and a state $s \in \mathcal{S}_h$ such that*

$$\eta(s) > \sup_{\pi_{mix} \in \Delta(\{\pi^\star_s\}_{s \in \mathcal{S}_{h-1}}), \pi_h \in \Upsilon} \mathbb{P}_{\pi_{min} \circ_h \pi_h}(s).$$

Here π_{mix} is a mixture policy *over the homing policies $\{\pi^\star_s\}_{s \in \mathcal{S}_{h-1}}$ for the states at time step $h-1$, and $\circ_h$ denotes policy composition at time h.*

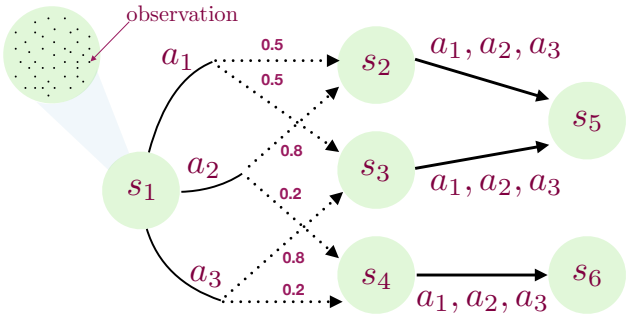


Figure 4: A Block MDP example showing non-compositional nature of homing policies. The Block MDP has six states $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ and three actions $\mathcal{A} = \{a_1, a_2, a_3\}$. The agent starts deterministically in s_1. Dashed lines denote stochastic transitions while solid lines are deterministic. The numbers on each dashed arrow depict the transition probabilities. We do not show observations for every state for brevity.

Proof. See Figure 4. The homing policy for s_5 takes action a_1 in s_1, which yields a visitation probability for s_5 of 1. However, the homing policies for states s_2, s_3, and s_4 do not take action a_1 in s_1. □

B. Properties of Kinematic Inseparability

In this section, we establish several useful properties for kinematically inseparable (KI) state abstractions. We recall the definition of forward kinematic inseparability, backward kinematic inseparability, and kinematic inseparability below.

Definition 5 (Forward Kinematic Inseparability)**.** *Two observations $x_1, x_2 \in \mathcal{X}$ are* forward kinematically inseparable *(KI) if for every $x' \in \mathcal{X}$ and $a \in \mathcal{A}$ we have $T(x' \mid x_1, a) = T(x' \mid x_2, a)$.*

Definition 6 (Backward Kinematic Inseparability)**.** *Two observations $x'_1, x'_2 \in \mathcal{X}$ are* backward kinematically inseparable *(KI) if for all distributions $u \in \Delta(\mathcal{X} \times \mathcal{A})$ supported on $\mathcal{X} \times \mathcal{A}$ and $\forall x \in \mathcal{X}, a \in \mathcal{A}$ we have*

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

$\mathbb{P}_u(x, a \mid x')$ 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.*

Definition 7 (Kinematic Inseparability)**.** *Two observations x'_1, x'_2 are* kinematically inseparable *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 $a, a' \in \mathcal{A}$ we have*

$$\mathbb{P}_u(x, a \mid x'_1) = \mathbb{P}_u(x, a \mid x'_2) \quad \textit{and} \quad T(x'' \mid x'_1, a') = T(x'' \mid x'_2, a').$$

Fact 2. *Forward kinematic inseparability (KI), backward KI and KI defines an equivalence relation on $\mathcal{X}$.*

Proof. That these relations are reflexive, symmetric, and transitive, all follow trivially from the definitions, in particular using the fact that equality itself is symmetric and transitive. □

Lemma 4. *Let $x_1, x_2 \in \mathcal{X}$. If $g^\star(x_1) = g^\star(x_2)$ then x_1 and x_2 are KI. This implies that they are also forward KI and backward KI.*

Proof. Fix any $x \in \mathcal{X}$, $a \in \mathcal{A}$ and $u \in \Delta(\mathcal{X} \times \mathcal{A})$ with $\text{supp}(u) = \mathcal{X} \times \mathcal{A}$. We show below that x_1 and x_2 are forward KI and backward KI which together establishes that desired claim.

Forward KI: By the Block MDP structure, we have

$$T(x \mid x_1, a) = T(x \mid g^\star(x_1), a) = T(x \mid g^\star(x_2), a) = T(x \mid x_2, a)$$

Backward KI: Again, using the Block MDP structure:

$$\begin{aligned}\mathbb{P}_u(x,a \mid x_1) &= \frac{T(x_1 \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[T(x_1 \mid \tilde{x},\tilde{a})\right]} = \frac{q(x_1 \mid g^\star(x_1))T(g^\star(x_1) \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[q(x_1 \mid g^\star(x_1))T(g^\star(x_1) \mid \tilde{x},\tilde{a})\right]} \\ &= \frac{T(g^\star(x_1) \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[T(g^\star(x_1) \mid \tilde{x},\tilde{a})\right]} = \frac{T(g^\star(x_2) \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[T(g^\star(x_2) \mid \tilde{x},\tilde{a})\right]} \\ &= \frac{q(x_2 \mid g^\star(x_2))T(g^\star(x_1) \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[q(x_2 \mid g^\star(x_2))T(g^\star(x_1) \mid \tilde{x},\tilde{a})\right]} = \frac{T(x_2 \mid x,a)u(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim u}\left[T(x_2 \mid \tilde{x},\tilde{a})\right]} = \mathbb{P}_u(x,a \mid x_2). \qquad \square\end{aligned}$$

The next simple fact shows that observations that appear at different time points are always separable.

Fact 3. *If x, x' are forward or backward KI, then $\tau(x) = \tau(x')$, where recall that $\tau(x)$ denotes the time step where x is reachable.*

Proof. If $h := \tau(x) \neq \tau(x') := h'$ then $T(\cdot \mid x,a) \in \Delta(\mathcal{X}_{h+1})$ while $T(\cdot \mid x',a) \in \Delta(\mathcal{X}_{h'+1})$, so these distributions cannot be equal. A similar argument holds for Backward KI. $\square$

Using the transitivity property for backward KI, we can consider sets of observations that are all pairwise backward KI. The next lemma provides a convenient characterization for backward KI sets.

Lemma 5. *Let $\mathcal{X}' \subseteq \mathcal{X}$ be a set of backward KI observations. Then $\exists u \in \Delta(\mathcal{X})$ with supp$(u) = \mathcal{X}$ such that for all $x', x'' \in \mathcal{X}'$ we have:*

$$\forall x \in \mathcal{X}, a \in \mathcal{A}, \qquad \frac{T(x' \mid x,a)}{u(x')} = \frac{T(x'' \mid x,a)}{u(x'')}. \tag{1}$$

The converse is also true: if (1) holds for some $u \in \Delta(\mathcal{X})$ with full support and all $x', x'' \in \mathcal{X}' \subset \mathcal{X}$ then $\mathcal{X}'$ are is a backward KI set.

Proof. Fix $\tilde{u} \in \Delta(\mathcal{X} \times \mathcal{A})$ with supp$(\tilde{u}) = \mathcal{X} \times \mathcal{A}$. Define $u(x) := \mathbb{E}_{(\tilde{x},\tilde{a})\sim\tilde{u}}\left[T(x \mid \tilde{x},\tilde{a})\right]$. Observe that by construction $u(x) > 0$ for all $x \in \mathcal{X}$. Let $x', x'' \in \mathcal{X}'$ then as x' and x'' are backward KI, we have that for all $x \in \mathcal{X}, a \in \mathcal{A}$:

$$\begin{aligned}\mathbb{P}_{\tilde{u}}(x,a \mid x'') = \mathbb{P}_{\tilde{u}}(x,a \mid x') &\Rightarrow \frac{T(x'' \mid x,a)\tilde{u}(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim\tilde{u}}\left[T(x'' \mid \tilde{x},\tilde{a})\right]} = \frac{T(x' \mid x,a)\tilde{u}(x,a)}{\mathbb{E}_{(\tilde{x},\tilde{a})\sim\tilde{u}}\left[T(x' \mid \tilde{x},\tilde{a})\right]} \\ &\Rightarrow \frac{T(x'' \mid x,a)}{u(x'')} = \frac{T(x' \mid x,a)}{u(x')}.\end{aligned}$$

For the converse, let $\tilde{u} \in \Delta(\mathcal{X} \times \mathcal{A})$ have full support. Then we have

$$\mathbb{P}_{\tilde{u}}(x,a \mid x_1') = \frac{T(x_1' \mid x,a)\tilde{u}(x,a)}{\sum_{\tilde{x},\tilde{a}} T(x_1' \mid \tilde{x},\tilde{a})\tilde{u}(\tilde{x},\tilde{a})} = \frac{\frac{u(x_1')}{u(x_2')}T(x_2' \mid x,a)\tilde{u}(x,a)}{\sum_{\tilde{x},\tilde{a}} \frac{u(x_1')}{u(x_2')}T(x_2' \mid \tilde{x},\tilde{a})\tilde{u}(\tilde{x},\tilde{a})} = \mathbb{P}_{\tilde{u}}(x,a \mid x_2'),$$

and so x_1, x_2 are backward KI. $\square$

We next show that an ordering relation between policy visitation probabilities is preserved through backward KI. This key structural property allows us to use the backward KI relationship to find a policy cover.

Lemma 6. *Let $\mathcal{X}' \subseteq \mathcal{X}$ be a set of backward KI observations and let $\mathcal{X}_1', \mathcal{X}_2' \subset \mathcal{X}'$. For any $\pi_1, \pi_2 \in \Upsilon$, we have $\frac{\mathbb{P}_{\pi_1}(\mathcal{X}_1')}{\mathbb{P}_{\pi_2}(\mathcal{X}_1')} = \frac{\mathbb{P}_{\pi_1}(\mathcal{X}_2')}{\mathbb{P}_{\pi_2}(\mathcal{X}_2')}$.*

Proof. Assume that $\mathcal{X}' \subseteq \mathcal{X}_h$ for some h, which is without loss of generality, since if they are observable at different time steps, then they are trivially separable. From Lemma 5, there exists a $u \in \Delta(\mathcal{X})$ supported everywhere such that for any $x_1', x_2' \in \mathcal{X}'$ we have: $\frac{T(x_1' \mid x,a)}{u(x_1')} = \frac{T(x_2' \mid x,a)}{u(x_2')}$ for any $x \in \mathcal{X}$ and $a \in \mathcal{A}$. Let π be any policy and define its occupancy

measure at time $h-1$, $\xi_{h-1} \in \Delta(\mathcal{X}_{h-1} \times \mathcal{A})$, as $\xi_{h-1}(x,a) := \mathbb{E}_\pi\left[\mathbf{1}\{x_{h-1} = x, a_{h-1} = a\}\right]$. Then for any fixed $\tilde{x} \in \mathcal{X}'$ and $j \in \{1,2\}$ we have

$$\mathbb{P}_\pi(\mathcal{X}'_j) = \mathbb{E}_{(x,a)\sim\xi}\left[\sum_{x'\in\mathcal{X}'_j} T(x' \mid x, a)\right] = \mathbb{E}_{x,a\sim\xi}\left[\sum_{x'\in\mathcal{X}'_j} \frac{u(x')}{u(\tilde{x})} T(\tilde{x} \mid x, a)\right] = \frac{u(\mathcal{X}'_j)}{u(\tilde{x})}\mathbb{P}_\pi(\tilde{x}),$$

where the second inequality follows from Lemma 5. Re-arranging, we have that $\frac{\mathbb{P}_\pi(\mathcal{X}'_1)}{\mathbb{P}_\pi(\mathcal{X}'_2)} = \frac{u(\mathcal{X}'_1)}{u(\mathcal{X}'_2)}$, and as the right hand side does not depend on π, the result follows. □

Lastly, we show that a set of observations are backward KI, then a single policy simultaneously maximizes the visitation probability to these observations. Moreover, we can construct a reward function for which this common policy is the reward maximizer. Recall that Υ is the (unrestricted) set of *all* policies.

Lemma 7. *Let $\mathcal{X}' \subseteq \mathcal{X}$ be a set of backward kinematically inseparable observations. Then there exists a policy $\pi \in \Upsilon$ that maximizes $\mathbb{P}_\pi(x')$ simultaneously for all $x' \in \mathcal{X}'$. Further, this policy is the optimal policy for the internal reward function $R'(x,a) := 1\{x \in \mathcal{X}'\}$.*

Proof. Let $x_1, x_2 \in \mathcal{X}'$ and define $\pi := \operatorname{argmax}_{\pi\in\Upsilon} \mathbb{P}_\pi(x_1)$. Let $\pi_2 \in \Upsilon$ be any other policy. Then, by Lemma 6, we have

$$\mathbb{P}_\pi(x_1) \geq \mathbb{P}_{\pi_2}(x_1) \Leftrightarrow \mathbb{P}_\pi(x_2) \geq \mathbb{P}_{\pi_2}(x_2).$$

As the left hand side is true by definition of π, we see that π also maximizes the visitation probability for x_2. As this is true for any x_2, we have that π simultaneously maximizes the visitation probability for all $x \in \mathcal{X}'$.

Clearly, for this policy and the specified reward function R', we have

$$\mathbb{E}_\pi\left[\sum_h R'(x_h, a_h)\right] = \mathbb{P}_\pi(\mathcal{X}') = \sum_{x'\in\mathcal{X}'} \mathbb{P}_\pi(x') \geq \max_{\pi'\in\Upsilon} \sum_{x'\in\mathcal{X}'} \mathbb{P}_{\pi'}(x') = \max_{\pi'\in\Upsilon} \mathbb{E}_{\pi'}\left[\sum_h R'(x_h, a_h)\right].$$

Here we are assuming $\mathcal{X}$ is countable, as we have mentioned. □

C. Analysis of Policy Search by Dynamic Programming

This section provides a detailed statistical and computation analysis of the Policy Search by Dynamic Programming (PSDP) algorithm, with pseudocode in Algorithm 2. The main guarantee is as follows:

Theorem 4. *Let $(\hat{\pi}_1, \hat{\pi}_2, \cdots, \hat{\pi}_h) = \texttt{PSDP}(\Psi, R, h, \Pi, n)$ be the policy returned by Algorithm 2 using policy covers $\Psi = \{\Psi_t\}_{t=1}^h$ where Ψ_t is an α-policy cover for $\mathcal{S}_t$ and $|\Psi_t| \leq N$ for all $t \in [h]$. Assume that either R is an internal reward function corresponding to time $h+1$, or that R is the external reward function and $h = H$, and that Assumption 2 holds. Then for any $\delta \in (0,1)$, with probability at least $1 - h\delta$ we have:*

$$V(\hat{\pi}_{1:h}; R) \geq \max_{\pi_1,\ldots,\pi_h\in\Pi} V(\pi_{1:h}; R) - \frac{Nh\Delta_{csc}}{\alpha} \qquad \textit{where} \qquad \Delta_{csc} := 4\sqrt{\frac{|\mathcal{A}|}{n}\ln\left(\frac{2|\Pi|}{\delta}\right)}.$$

The algorithm runs in polynomial time with h calls to the contextual bandit oracle.

Before turning to the proof, we state a standard generalization bound for the contextual bandit problems created by the algorithm. These problems are induced by an underlying distribution Q over tuples $(x, \vec{r})$ where $x \in \mathcal{X}$ and $\vec{r} \in [0,1]^{|\mathcal{A}|}$, and a *logging policy* $\pi_{\log}$. Formally, we obtain tuples $(x, a, p, r) \sim Q_{\log}$ where $(x, \vec{r}) \sim Q$, $a \sim \pi_{\log}(x)$, $p := \pi_{\log}(a \mid x)$ is the probability of choosing the action for the current observation, and $r := \vec{r}(a)$. In our application, we always have $\pi_{\log} := \texttt{Unf}(\mathcal{A})$ so that $p = 1/|\mathcal{A}|$. Given a dataset of n tuples $D := \{(x_i, a_i, p_i, r_i)\}_{i=1}^n \overset{iid}{\sim} Q_{\log}$, we invoke the contextual bandit oracle, $\texttt{CB}(D, \Pi)$, to find a policy $\hat{\pi}$. The following proposition provides a performance guarantee for $\hat{\pi}$.

Proposition 5. *Let $D := \{(x_i, a_i, p_i, r_i)\}_{i=1}^n \overset{iid}{\sim} Q_{log}$ be a dataset of n samples from a contextual bandit distribution Q_{log} induced by the uniform logging policy interacting with an underlying distribution Q. Let $\hat{\pi} = \texttt{CB}(D, \Pi)$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\mathbb{E}_{(x,\vec{r})\sim Q}[\vec{r}(\hat{\pi}(x)] \geq \max_{\pi\in\Pi} \mathbb{E}_{(x,\vec{r})\sim Q}[\vec{r}(\pi(x)] - \Delta_{csc}.$$

Proof. The proof is a standard generalization bound for contextual bandits (c.f., Langford & Zhang, 2008). We provide a short proof for completeness.

For policy π, define $R(\pi) := \mathbb{E}_Q\left[\vec{r}(\pi(x))\right]$, $\hat{r}_i(\pi) = \frac{\mathbf{1}\{a_i=\pi(x_i)\}r_i}{p_i} = |\mathcal{A}|\mathbf{1}\{a_i = \pi(x_i)\}r_i$, and observe that the contextual bandit oracle finds the policy that maximizes $\hat{R}(\pi) := \frac{1}{n}\sum_{i=1}^n \hat{r}_i(\pi)$. The random variables $\hat{r}_i(\pi)$ satisfy the following useful properties:

$$\text{Unbiased: } \mathbb{E}_{Q_{\log}}\left[\hat{r}(\pi)\right] = \mathbb{E}_Q\left[\sum_a \pi_{\log}(a \mid x)\frac{\mathbf{1}\{a=\pi(x)\}\vec{r}(a)}{\pi_{\log}(a \mid x)}\right] = \mathbb{E}_Q\left[\vec{r}(\pi(x))\right].$$

$$\text{Low variance: } \mathrm{Var}[\hat{r}(\pi)] \le \mathbb{E}_{Q_{\log}}\left[\hat{r}^2(\pi)\right] \le \mathbb{E}_{Q_{\log}}\left[\frac{\mathbf{1}\{a=\pi(x)\}}{p^2}\right] = |\mathcal{A}|\cdot\mathbb{P}_{Q_{\log}}[a=\pi(x)] = |\mathcal{A}|$$

$$\text{Range: } |\hat{r}(\pi)| \le |\mathcal{A}|.$$

Therefore, using Bernstein's inequality (Proposition 10) and union bound we have that for any $\delta \in (0,1)$, with probability at least $1-\delta$:

$$\forall \pi \in \Pi, \qquad \left|\hat{R}(\pi) - R(\pi)\right| \le \frac{2|\mathcal{A}|}{3n}\ln\left(\frac{2|\Pi|}{\delta}\right) + \sqrt{\frac{2|\mathcal{A}|}{n}\ln\left(\frac{2|\Pi|}{\delta}\right)} =: \Delta.$$

The contextual bandit oracle finds $\hat{\pi}$ that maximizes the empirical quantity $\hat{R}(\pi)$, so, by a standard generalization argument, we have

$$R(\hat{\pi}) \ge \hat{R}(\pi) - \Delta \ge \max_{\pi\in\Pi}\hat{R}(\pi) - \Delta \ge \max_{\pi\in\Pi} R(\pi) - 2\Delta.$$

Of course as the reward vector is bounded in $[0,1]$ we always have $R(\hat{\pi}) \ge \max_{\pi\in\Pi} R(\pi) - 1$, which means that with probability at least $1-\delta$, we have

$$R(\hat{\pi}) \ge \max_{\pi\in\Pi} R(\pi) - \min\left\{1, 2\Delta\right\}.$$

Finally, if $2\Delta \le 1$ then $\frac{4|\mathcal{A}|\ln(2|\Pi|/\delta)}{3n} \le \sqrt{\frac{4|\mathcal{A}|\ln(2|\Pi|/\delta)}{3n}}$. This observation leads to the definition Δ_{csc}. □

Proof of Theorem 4. Let $\pi_1^\star, \pi_2^\star, \cdots \pi_h^\star = \mathrm{argmax}_{\pi_1,\pi_2,\cdots\pi_h\in\Pi} V(\pi_{1:h}; R)$ be the optimal non-stationary policy, for reward function R with time horizon h. PSDP solves a sequence of h contextual bandit problems to learn policies $\hat{\pi}_t$ for $t = h, \ldots, 1$. The t^{th} problem is induced by a distribution Q_t supported over $\mathcal{X}_t \times [0,1]^{|\mathcal{A}|}$, which is defined inductively as follows: The observations are induced by choosing a $\pi_t \sim \mathtt{Unf}(\Psi_t)$ and executing π_t for $t-1$ steps to visit x_t. The reward given x_t and an action $a \in \mathcal{A}$ is $R(x_{h+1})$ where the trajectory is completed by first executing a from x_t and then following $\hat{\pi}_{t+1:h}$. As in Proposition 5, the contextual bandit dataset is induced by this distribution Q_t the uniform logging policy.

By Proposition 5 with probability at least $1-h\delta$, we have that for all t, $\hat{\pi}_t$ satisfies

$$\mathbb{E}_{(x,\vec{r})\sim Q_t}\left[\vec{r}(\hat{\pi}_t(x))\right] \ge \max_{\pi\in\Pi}\mathbb{E}_{(x,\vec{r})\sim Q_t}\left[\vec{r}(\pi_t(x))\right] - \Delta_{csc},$$

where Q_t is as defined above. Using the definition of Q_t, this guarantee may be written as

$$\forall t \in [h] : \mathbb{E}_{x\sim Q_t}\left[V(x; \hat{\pi}_{t:h}, R)\right] \ge \max_{\pi\in\Pi}\mathbb{E}_{x\sim Q_t}\left[V(x; \pi\circ\hat{\pi}_{t+1:h}, R)\right] - \Delta_{csc}.$$

Define $Q_t^\star \in \Delta(\mathcal{X}_t)$ to be the distribution of observations visited by executing $\pi_{1:t-1}^\star$. By the performance difference lemma (Lemma 22) [c.f., Bagnell et al. (2004); Kakade (2003); Ross & Bagnell (2014)], we have

$$\begin{aligned} V(\hat{\pi}_{1:h}; R) - V(\pi_{1:h}^\star; R) &= \sum_{t=1}^h \mathbb{E}_{x_t\sim Q_t^\star}\left[V(x_t; \pi_t^\star\circ\hat{\pi}_{t+1:h}, R) - V(x_t; \hat{\pi}_{t:h}, R)\right] \\ &\le \sum_{t=1}^h \mathbb{E}_{x_t\sim Q_t^\star}\left[V(x_t; \tilde{\pi}_t^\star\circ\hat{\pi}_{t+1:h}, R) - V(x_t; \hat{\pi}_{t:h}, R)\right], \end{aligned}$$

where $\tilde{\pi}_t^\star(x) := \operatorname{argmax}_{a\in\mathcal{A}} \mathbb{E}\left[R(x,a) + V(x_{t+1}; \hat{\pi}_{t+1:h}, R) \mid x_t = x, a_t = a\right]$, for all $x \in \mathcal{X}_t$. With this definition, the inequality here is immediate, by definition of the value function.

Assumption 2 implies that $\tilde{\pi}_t^\star \in \Pi$ for each t, which is immediate for the external reward function. If we are using the internal reward function with some $h < H$, then by construction the internal reward function is defined only at time $h+1$, so we may simply append arbitrary policies $\hat{\pi}_{h+1:H}$ without affecting the reward or the value function. Formally, we have

$$\begin{aligned}
V(\hat{\pi}_{1:h}; R) - V(\pi^\star_{1:h}; R) &\leq \sum_{t=1}^{h} \mathbb{E}_{x_t\sim Q_t^\star}\left[V(x_t; \tilde{\pi}_t^\star \circ \hat{\pi}_{t+1:H}, R) - V(x_t; \hat{\pi}_{t:H}, R)\right] \\
&= \sum_{t=1}^{h} \mathbb{E}_{x_t\sim Q_t}\left[\frac{Q_t^\star(x_t)}{Q_t(x_t)}\left(V(x_t; \tilde{\pi}_t^\star \circ \hat{\pi}_{t+1:H}, R) - V(x_t; \hat{\pi}_{t:H}, R)\right)\right] \\
&\leq \sum_{t=1}^{h} \sup_{x_t}\left|\frac{Q_t^\star(x_t)}{Q_t(x_t)}\right| \cdot \mathbb{E}_{x_t\sim Q_t}\left[\left|V(x_t; \tilde{\pi}_t^\star \circ \hat{\pi}_{t+1:H}, R) - V(x_t; \hat{\pi}_{t:H}, R)\right|\right] \\
&\leq \sum_{t=1}^{h} \sup_{x_t}\left|\frac{Q_t^\star(x_t)}{Q_t(x_t)}\right| \cdot \Delta_{csc}.
\end{aligned}$$

The first line appends $\hat{\pi}_{h+1:H}$ to the roll-out policy, which as we argued does not affect the value function for any policy. The second line simply introduces the distribution Q_t that we used for learning $\hat{\pi}_t$. The third line is Holder's inequality, and in the fourth line, we use the fact that $\tilde{\pi}_t^\star$ is *pointwise* better than $\hat{\pi}_t$, so we can remove the absolute values. Then we simply use our guarantee from Proposition 5.

We finish the proof by using the policy cover property (Definition 2), namely that

$$\sup_{x_t}\left|\frac{Q_t^\star(x_t)}{Q_t(x_t)}\right| = \sup_{s_t}\left|\frac{\mathbb{P}_{\pi^\star_{1:t-1}}[s_t]}{\frac{1}{|\Psi_t|}\sum_{\pi\in\Psi_t}\mathbb{P}_\pi[s_t]}\right| \leq \sup_{s_t}\frac{\eta(s_t)}{\frac{1}{N}\alpha\eta(s_t)} = \frac{N}{\alpha}.$$

Combining terms proves the theorem. □

D. Analysis for the HOMER algorithm

In this section we present the analysis for HOMER. The proof is inductive in nature, proceeding from time point $h = 1$ to $h = H$, where for the h^{th} step, we use the policy covers from time points $h' = 1, \ldots, h-1$ to construct the policy cover at time h. Formally, the inductive claim is that for each h, given α-policy covers $\Psi_1, \ldots, \Psi_{h-1}$ over $\mathcal{S}_1, \ldots, \mathcal{S}_{h-1}$, the h^{th} iteration of HOMER finds an α-policy cover Ψ_h for $\mathcal{S}_h$. We will verify the base case shortly, and we break down the inductive argument into two main components: In the first part, we analyze the contrastive estimation problem and introduce a coupling to show that the supervised learning problem relates to backward kinematic inseparability. In this second part, we use this coupling to show that invoking PSDP with the learned representation yields a policy cover for time point h.

The base case. The base case is that Ψ_1 found by the algorithm is a policy cover over $\mathcal{S}_1$. This is easy to see, since for any states $s \in \mathcal{S}_1$, we have $\eta(s) = \mu(s)$, where recall that μ is the starting stat distribution. We can define Ψ_1 to be any finite set of policies, which immediately is a 1-policy cover, but since we never actually execute these policies, we simply set $\Psi_1 = \emptyset$ in Algorithm 1 (line 1).

D.1. The supervised learning problem and a coupling

In this subsection we analyze the supervised learning problem induced by HOMER, which is a form of contrastive estimation (line 11 in Algorithm 1). We reason about the Bayes optimal predictor for this problem, obtain a finite-sample generalization bound, and introduce a coupling to elucidate the connection to backward kinematic inseparability. Fix a time point $h \in \{2, \ldots, H\}$ and inductively assume that we have α policy covers $\Psi_1, \ldots, \Psi_{h-1}$ over $\mathcal{S}_1, \ldots, \mathcal{S}_{h-1}$ respectively. For the rest of this subsection, we suppress dependence on h in observations, that is we will always take $x \in \mathcal{X}_{h-1}$, and $x' \in \mathcal{X}_h$.

The supervised learning problem at time h is induced by a distribution $D \in \Delta(\mathcal{X}_{h-1}, \mathcal{A}, \mathcal{X}_h, \{0,1\})$, which is defined as follows: Two tuples are obtained $(x_1, a_1, x_1'), (x_2, a_2, x_2')$ are obtained by sampling $\pi_1, \pi_2 \sim \texttt{Unf}(\Psi_{h-1})$, and executing

the policies $\pi_1 \circ \texttt{Unf}(\mathcal{A})$ and $\pi_2 \circ \texttt{Unf}(\mathcal{A})$, respectively. Then with probability $1/2$ the sample from D is $(x_1, a_1, x_1', 1)$ and with the remaining probability, the sample is $(x_1, a_1, x_2', 0)$. Let $D(x, a, x' \mid y)$ be the conditional probability of the triple, conditional on the label y. Let $\rho_h \in \Delta(\mathcal{X}_h)$ denote the marginal probability distribution over x', that is $\rho_h(x') := \mathbb{P}_{\pi \circ \texttt{Unf}:\pi \sim \texttt{Unf}(\Psi_{h-1})}[x']$, and let $\mu_{h-1} \in \Delta(\mathcal{X}_{h-1})$ be the marginal distribution over x, that is $\mu_{h-1}(x) := \mathbb{P}_{\pi \sim \texttt{Unf}(\Psi_{h-1})}[x]$. These definitions are lifted to the state spaces $\mathcal{S}_{h-1}$ and $\mathcal{S}_h$ in the natural way.

With these definition, we have

$$D(x, a, x' \mid y = 1) = \frac{\mu_{h-1}(x)}{|\mathcal{A}|} \cdot T(x' \mid x, a), \qquad D(x, a, x' \mid y = 0) = \frac{\mu_{h-1}(x)}{|\mathcal{A}|} \cdot \rho_h(x').$$

The first lemma uses the fact that Ψ_{h-1} is an α-policy cover to lower bound the marginal probability $\rho_h(s_h)$, which ensure we have adequate coverage in our supervised learning problem.

Lemma 8. *If Ψ_{h-1} is an α-policy cover over $\mathcal{S}_{h-1}$, then for any $s \in \mathcal{S}_h$, we have $\rho_h(s) \geq \frac{\alpha \eta(s)}{N|\mathcal{A}|}$.*

Proof. For any $s \in \mathcal{S}_h$, we first upper bound $\eta(s)$ by

$$\begin{aligned}
\eta(s) &= \sup_{\pi \in \Pi_{\mathrm{NS}}} \mathbb{P}_\pi(s) = \sup_{\pi \in \Pi_{\mathrm{NS}}} \sum_{s_{h-1} \in \mathcal{S}_{h-1}} \mathbb{P}_\pi[s_{h-1}] \mathbb{E}_{x \sim q(.|s_{h-1})} \left[\sum_{a \in \mathcal{A}} \pi(a \mid x) T(s \mid s_{h-1}, a) \right] \\
&\leq \sum_{s_{h-1} \in \mathcal{S}_{h-1}} \sup_{\pi \in \Pi_{\mathrm{NS}}} \mathbb{P}_\pi[s_{h-1}] \sum_{a \in \mathcal{A}} T(s \mid s_{h-1}, a) = \sum_{s_{h-1} \in \mathcal{S}_{h-1}} \eta(s_{h-1}) \sum_{a \in \mathcal{A}} T(s \mid s_{h-1}, a).
\end{aligned}$$

We can also lower bound ρ_h as

$$\rho_h(s) = \sum_{\substack{s_{h-1} \in \mathcal{S}_{h-1} \\ a \in \mathcal{A}}} \frac{\mathbb{P}_{\pi \sim \texttt{Unf}(\Psi_{h-1})}[s_{h-1}]}{|\mathcal{A}|} T(s \mid s_{h-1}, a) \geq \frac{\alpha}{N|\mathcal{A}|} \sum_{\substack{s_{h-1} \in \mathcal{S}_{h-1} \\ a \in \mathcal{A}}} \eta(s_{h-1}) T(s \mid s_{h-1}, a) \geq \frac{\alpha \eta(s)}{N|\mathcal{A}|}.$$

Here the first identity expands the definition, and in the first inequality we use the fact that Ψ_{h-1} is an α-policy cover. The last inequality uses our upper bound on $\eta(s)$. □

The next lemma characterizes the Bayes optimal predictor for square loss minimization with respect to D. Recall that the Bayes optimal classifier is defined as

$$f^\star := \operatorname*{argmin}_f \mathbb{E}_{(x,a,x',y) \sim D} \left[\left(f(x, a, x') - y \right)^2 \right]$$

where the minimization is over *all* measurable functions.

Lemma 9. *The Bayes optimal predictor for square loss minimization over D is*

$$f^\star(x, a, x') := \frac{T(g^\star(x') \mid g^\star(x), a)}{T(g^\star(x') \mid g^\star(x), a) + \rho_h(g^\star(x'))}$$

Under Assumption 2, we have that $f^\star \in \mathcal{F}_N$ for any $N \geq N_{KD}$.

Proof. As we are using the square loss, the Bayes optimal predictor is the conditional mean, so $f^\star(x, a, x') = \mathbb{E}_D[y \mid (x, a, x')] = D(y = 1 \mid x, a, x')$. By Bayes rule and the fact that the marginal probability for both labels is $1/2$, we have

$$\begin{aligned}
D(y = 1 \mid x, a, x') &= \frac{D(x, a, x' \mid y = 1)}{D(x, a, x' \mid y = 1) + D(x, a, x' \mid y = 0)} = \frac{T(x' \mid x, a)}{T(x' \mid x, a) + \rho_h(x')} \\
&= \frac{T(g^\star(x') \mid g^\star(x), a)}{T(g^\star(x') \mid g^\star(x), a) + \rho_h(g^\star(x'))}.
\end{aligned}$$

□

Now that we have characterized the Bayes optimal predictor, we turn to the learning rule. We perform empirical risk minimization over n iid samples from D to learn a predictor $\hat{f} \in \mathcal{F}_N$ (We will bind $n = n_{\mathrm{reg}}$ toward the end of the proof). As $\mathcal{F}_N$ has pointwise metric entropy growth rate $\ln \mathcal{N}(\mathcal{F}_N, \varepsilon) \leq c_0 d_N \ln(1/\varepsilon)$, a standard square loss generalization analysis (see Proposition 11) yields the following corollary, which follows easily from Proposition 11.

Corollary 6. *For any $\delta \in (0,1)$ with probability at least $1-\delta$, the empirical risk minimizer, $\hat{f}$ based n iid samples from D satisfies*[4]

$$\mathbb{E}_D\left[\left(\hat{f}(x,a,x') - f^\star(x,a,x')\right)^2\right] \le \Delta_{reg} \text{ with } \Delta_{reg} := \frac{16\left(\ln|\Phi_N| + N^2|\mathcal{A}|\ln(n) + \ln(2/\delta)\right)}{n}.$$

Proof. The proof follows from a bound on the pointwise covering number of the class $\mathcal{F}_N$. For any $\varepsilon > 0$ we first form a cover of the class $\mathcal{W}_N$ by discretizing the output space to $Z := \{\epsilon, \dots, \lfloor 1/\epsilon \rfloor \epsilon\}$, and letting W_N be all functions from $[N] \times \mathcal{A} \times [N] \to Z$. Clearly we have $|W_N| \le (1/\epsilon)^{N^2|\mathcal{A}|}$, and it is easy to see that W_N is a pointwise cover for $\mathcal{W}_N$. Then we form $F_N = \{(x,a,x') \mapsto w(\phi(x),a,\phi'(x')) : w \in W_N, \phi, \phi' \in \Phi_N\}$, which is clearly a pointwise cover for $\mathcal{F}_N$ and has size $|\Phi_N|^2|W_N|$. In other words, the pointwise log-covering number is $N^2|\mathcal{A}|\ln(1/\varepsilon) + 2\ln|\Phi_N|$, which we plug into the bound in Proposition 11. Taking $\varepsilon = 1/n$ there the bound from Proposition 11 is at most

$$\frac{6}{n} + \frac{16\ln|\Phi_N| + 8N^2|\mathcal{A}|\ln(n) + 8\ln(2/\delta)}{n} \le \Delta_{reg}. \qquad \square$$

The coupling. For the next step of the proof, we introduce a useful coupling distribution based on D. Let $D_{\text{coup}} \in \Delta(\mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h \times \mathcal{X}_h)$ have density $D_{\text{coup}}(x,a,x'_1,x'_2) = D(x,a)\rho_h(x'_1)\rho_h(x'_2)$. That is, we sample x, a by choosing $\pi \sim \texttt{Unf}(\Psi_{h-1})$, rolling in, and then taking a uniform action $a_{h-1} \sim \texttt{Unf}(\mathcal{A})$. Then, we obtain x'_1, x'_2 independently by sampling from the marginal distribution ρ_h induced by D.

It is also helpful to define the shorthand notation $V : \mathcal{X}_h \times \mathcal{X}_h \times \mathcal{X}_{h-1} \times \mathcal{A} \to \mathbb{R}$ by

$$V(x'_1,x'_2,x,a) := \frac{T(g^\star(x'_1) \mid g^\star(x),a)}{\rho_h(g^\star(x'_1))} - \frac{T(g^\star(x'_2) \mid g^\star(x),a)}{\rho_h(g^\star(x'_2))}.$$

This function is lifted to operate on states $\mathcal{S}_h$ in the natural way. Note also that, as $\rho_h(\cdot) > 0$ everywhere, V is well-defined. Observe that V is related to the notion of backward kinematic inseparability. Finally, define

$$b_i := \mathbb{E}_{x' \sim \rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x') = i\}\right],$$

which is the prior probability over the learned abstract state i, where $\hat{\phi}_h^{(\mathrm{B})}$ is the learned abstraction function for time h implicit in the predictor $\hat{f}$. In the next lemma, we show that $\hat{\phi}_h^{(\mathrm{B})}$ approximately learns a backward KI abstraction by relating the excess risk of $\hat{f}$ to the performance of the decoder via the V function.

Lemma 10. *Let $\hat{f} =: (\hat{w}, \hat{\phi}_{h-1}^{(\mathrm{F})}, \hat{\phi}_h^{(\mathrm{B})})$ be the empirical risk minimizer on n iid samples from D, that is the output of $\texttt{REG}(\mathcal{F}_N, \mathcal{D})$. Under the $1-\delta$ event of Corollary 6, for each $i \in [N]$ we have*

$$\mathbb{E}_{D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = i = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\,|V(x'_1,x'_2,x,a)|\right] \le 8\sqrt{b_i \Delta_{reg}}.$$

Proof. For the proof, it is helpful to introduce a second coupled distribution D'_{coup} in which x, a are sampled as before, but now $x'_1, x'_2 \overset{iid}{\sim} D(\cdot \mid x,a)$, instead of from the prior. Note that this condition probability is $D(x' \mid x,a) = 1/2q(x' \mid g^\star(x))\,(T(g^\star(x') \mid g^\star(x),a) + \rho_h(g^\star(x')))$. To translate from D_{coup} to D'_{coup} we expand the definition of V and introduce $f^\star$. The main observation here is that V is normalized by $\rho_h(\cdot)$ but $f^\star$ is normalized, essentially by $D(\cdot \mid x,a)$.

$$\begin{aligned} V(x'_1,x'_2,x,a) &= \frac{\rho_h(x'_2)T(x'_1 \mid x,a) - \rho_h(x'_1)T(x'_2 \mid x,a)}{\rho_h(x'_1)\rho_h(x'_2)} \\ &= \frac{4D(x'_1 \mid x,a)D(x'_2 \mid x,a)}{\rho_h(x'_1)\rho_h(x'_2)} \cdot \left(f^\star(x,a,x'_1) - f^\star(x,a,x'_2)\right). \end{aligned}$$

[4] As we remark in Appendix F, sharper generalization analyses are possible, with more refined notions of statistical complexity. Such results are entirely composable with our analyses.

The last step follows since, in the first term the emission distributions cancel, while the cross terms cancel when we introduce the least common multiple in the term $f^\star(x,a,x'_1) - f^\star(x,a,x'_2)$. Specifically

$$\begin{aligned} f^\star(x,a,x'_1) - f^\star(x,a,x'_2) &= \frac{T(g^\star(x'_1) \mid g^\star(x),a)}{T(g^\star(x'_1) \mid g^\star(x),a) + \rho_h(g^\star(x'_1))} - \frac{T(g^\star(x'_2) \mid g^\star(x),a)}{T(g^\star(x'_2) \mid g^\star(x),a) + \rho_h(g^\star(x'_2)} \\ &= \frac{\rho_h(g^\star(x'_2))T(g^\star(x'_1) \mid g^\star(x),a) - \rho_h(g^\star(x'_1))T(g^\star(x'_2) \mid g^\star(x),a)}{(T(g^\star(x'_1) \mid g^\star(x),a) + \rho_h(g^\star(x'_1)))\,T(g^\star(x'_2) \mid g^\star(x),a) + \rho_h(g^\star(x'_2))}. \end{aligned}$$

Nevertheless, this calculation lets us translate from D_{coup} to D'_{coup} while moving from V to $f^\star$. For shorthand, let $\mathcal{E}_i := \mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = i = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}$, so the above derivation yields

$$\mathbb{E}_{D_{\text{coup}}}\left[\mathcal{E}_i \cdot |V(x'_1,x'_2,x,a)|\right] = 4\mathbb{E}_{D'_{\text{coup}}}\left[\mathcal{E}_i \cdot |f^\star(x,a,x'_1) - f^\star(x,a,x'_2)|\right]. \tag{2}$$

Now that we have introduce $f^\star$, we can introduce $\hat{f}$ and relate to the excess risk

$$\begin{aligned} &\mathbb{E}_{D'_{\text{coup}}}\left[\mathcal{E}_i \cdot |f^\star(x,a,x'_1) - f^\star(x,a,x'_2)|\right] \\ &\quad\le \mathbb{E}_{D'_{\text{coup}}}\left[\mathcal{E}_i \cdot \left(\left|f^\star(x,a,x'_1) - \hat{f}(x,a,x'_1)\right| + \left|f^\star(x,a,x'_2) - \hat{f}(x,a,x'_1)\right|\right)\right] \\ &\quad= \mathbb{E}_{D'_{\text{coup}}}\left[\mathcal{E}_i \cdot \left(\left|f^\star(x,a,x'_1) - \hat{f}(x,a,x'_1)\right| + \left|f^\star(x,a,x'_2) - \hat{f}(x,a,x'_2)\right|\right)\right] \\ &\quad\le 2\mathbb{E}_D\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x') = i\}\left|f^\star(x,a,x') - \hat{f}(x,a,x')\right|\right] \le 2\sqrt{b_i \mathbb{E}_D\left[\left(f^\star(x,a,x') - \hat{f}(x,a,x')\right)^2\right]} \\ &\quad\le 2\sqrt{b_i \Delta_{reg}}. \end{aligned}$$

The first step is the triangle inequality. The key step is the second one: the identity uses the fact that under event $\mathcal{E}_i$, we know that $\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)$, which, by the bottleneck structure of $\hat{f}$, yields that $\hat{f}(x,a,x'_1) = \hat{f}(x,a,x'_2)$. In the third step we combine terms, drop the dependence on the other observaiton, and use the fact that D'_{coup} shares marginal distributions with D. Finally, we use the Cauchy-Schwarz inequality, the fact that $\mathbb{E}_D \mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x') = i\} = b_i$, and Corollary 6. Combining with (2) proves the lemma. □

An aside on realizability. Given the bottleneck structure of our function class $\mathcal{F}_N$, it is important to ask whether realizability is even feasible. In particular for a bottleneck capacity of N, each $f \in \mathcal{F}_N$ has a range of at most $N^2|\mathcal{A}|$ discrete values. If we choose N to be too small, we may not have enough degrees of freedom to express $f^\star$. By inspection $f^\star$ has a range of at most $|\mathcal{S}|^2|\mathcal{A}|$, so certainly a bottleneck capacity of $N \ge |\mathcal{S}|$ suffices. In the next proposition, we show that in fact $N \ge \mathrm{N_{KD}}$ suffices, which motivates the condition in Assumption 2.

Proposition 7. *Fix $h \in \{2,\dots,H\}$. If $x_1, x_1 \in \mathcal{X}_{h-1}$ are kinematically inseparable observations, then for all $x' \in \mathcal{X}_h$ and $a \in \mathcal{A}$, we have $f^\star(x_1,a,x') = f^\star(x_2,a,x')$. Analogously, if $x'_1, x'_2 \in \mathcal{X}_h$ are kinematically inseparable, then for all $x \in \mathcal{X}_{h-1}$ and $a \in \mathcal{A}$, we have $f^\star(x,a,x'_1) = f^\star(x,a,x'_2)$.*

Proof. We prove the forward direction first. As x_1, x_2 are forward KI, we have

$$f^\star(x_1,a,\tilde{x}) = \frac{T(\tilde{x} \mid x_1,a)}{T(\tilde{x} \mid x_1,a) + \rho_h(\tilde{x})} = \frac{T(\tilde{x} \mid x_2,a)}{T(\tilde{x} \mid x_2,a) + \rho_h(\tilde{x})} = f^\star(x_2,a,\tilde{x}).$$

For the backward direction, as x'_1, x'_2 are backward KI, from Lemma 5 there exists $u \in \Delta(\mathcal{X})$ with $\mathrm{supp}(u) = \mathcal{X}$ such that $\frac{T(x'_1|x,a)}{u(x'_1)} = \frac{T(x'_2|x,a)}{u(x'_2)}$. Further, ρ_h satisfies $\rho_h(x'_1) = \frac{u(x'_1)}{u(x'_2)}\rho_h(x'_2)$. Thus, we obtain

$$f^\star(x,a,x'_1) = \frac{T(x'_1 \mid x,a)}{T(x'_1 \mid x,a) + \rho_h(x'_1)} = \frac{\frac{u(x'_1)}{u(x'_2)}T(x'_2 \mid x,a)}{\frac{u(x'_1)}{u(x'_2)}T(x'_2 \mid x,a) + \frac{u(x'_1)}{u(x'_2)}\rho_h(x'_2)} = f^\star(x,a,x'_2). \qquad \square$$

D.2. Building the policy cover

Lemma 10 relates our learned decoder function $\hat{\phi}_h^{(\mathrm{B})}$ to backward KI. For some intuition as to why, it is helpful to consider the asymptotic regime, where $n \to \infty$, so that $\Delta_{reg} \to 0$. In this regime, Lemma 10 shows that whenever $\hat{\phi}_h^{(\mathrm{B})}$ maps two observations to the same abstract state, these observations must have $V = 0$ for all x, a. As our distribution has full support, by Lemma 5, these observations must be backward KI. Of course, this argument only applies to the asymptotic regime. In this section, we establish a finite-sample analog, and we show how using the internal reward functions induced by $\hat{\phi}_h^{(\mathrm{B})}$ in PSDP yields a policy cover for $\mathcal{S}_h$.

The first lemma is a comparison lemma, which lets us compare visitation probabilities for two policies. To state the lemma we will define a helpful quantity for any $s \in \mathcal{S}, i \in [N]$:

$$P_{s,i} := \mathbb{E}_{x' \sim \rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x') = i \wedge g^\star(x') = s\}\right]$$

Lemma 11. *Assume Ψ_{h-1} is an α policy cover for $\mathcal{S}_{h-1}$. Then for any two policies π_1, π_2 and any state $s \in \mathcal{S}_h$, we have*

$$\mathbb{P}[s \mid \pi_1] - \mathbb{P}[s \mid \pi_2] \leq \min_{i \in [N]} \left\{ \frac{\rho_h(s)}{b_i}\left(\mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i \mid \pi_1] - \mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i \mid \pi_2]\right) + \frac{16N|\mathcal{A}|\rho_h(s)}{\alpha P_{s,i} b_i^{1/2}}\sqrt{\Delta_{reg}} \right\}.$$

Proof. The key step is to observe that by the definition of V

$$\forall x_2', x, a : \mathbb{E}_{x_1' \sim \rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} V(x_1', x_2', x, a)\right] = \sum_{x_1'} \mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} T(x_1' \mid x, a) - \frac{b_i T(x_2' \mid x, a)}{\rho_h(x_2')}.$$

Using this identity, we may express the visitation probability for a policy π as

$$\begin{aligned}
\mathbb{P}[s \mid \pi] &= \mathbb{E}_{(x,a) \sim \pi} \sum_{x_2'} \mathbf{1}\{g^\star(x_2') = s\} T(x_2' \mid x, a) \\
&= \frac{1}{b_i}\mathbb{E}_{(x,a) \sim \pi, x_2' \sim \rho_h} \sum_{x_1'} \mathbf{1}\{g^\star(x_2') = s \wedge \hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} \left(T(x_1' \mid x, a) - \rho_h(x_1') V(x_1', x_2', x, a)\right) \\
&= \frac{\rho_h(s)}{b_i}\mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i \mid \pi] - \frac{1}{b_i}\mathbb{E}_{D_{\text{coup}}}\left[\frac{\pi(x,a)}{D(x,a)}\mathbf{1}\{g^\star(x_2') = s \wedge \hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} V(x_1', x_2', x, a)\right].
\end{aligned}$$

Here we are using the shorthand $\pi(x,a) = \mathbb{P}[x \mid \pi]\pi(a \mid x)$ for the policy occupancy measure, with a similar notation the distribution D induced by our policy cover Ψ_{h-1}. Using the inductive hypothesis that Ψ_{h-1} is a α-policy cover (essentially Lemma 8), we have

$$\left|\frac{\pi(x,a)}{D(x,a)}\right| = \left|\frac{\mathbb{P}[x \mid \pi] \cdot \pi(a \mid x)}{\mathbb{E}_{\pi' \sim \mathtt{Unf}(\Psi_{h-1})}\mathbb{P}[x \mid \pi'] \cdot 1/|\mathcal{A}|}\right| \leq |\mathcal{A}| \left|\frac{\mathbb{P}[g^\star(x) \mid \pi]}{\mathbb{E}_{\pi' \sim \mathtt{Unf}(\Psi_{h-1})}\mathbb{P}[g^\star(x) \mid \pi']}\right| \leq \frac{|\mathcal{A}|N}{\alpha}.$$

Combining, absolute value of the second term above is at most

$$\frac{N|\mathcal{A}|}{\alpha b_i}\mathbb{E}_{D_{\text{coup}}}\left[\mathbf{1}\{g^\star(x_2') = s \wedge \hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} |V(x_1', x_2', x, a)|\right].$$

Let us now work with just the expectation. Recall that we can lift the definition of V to operate on states $s \in \mathcal{S}_h$ in lieu of observations. Using this fact, we have that under the probability $1 - \delta$ event of Lemma 10

$$\begin{aligned}
&\mathbb{E}_{D_{\text{coup}}}\left[\mathbf{1}\{g^\star(x_2') = s \wedge \hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} |V(x_1', x_2', x, a)|\right] = \rho_h(s)\mathbb{E}_{(x,a) \sim D, x_1' \sim \rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} |V(x_1', s, x, a)|\right] \\
&= \rho_h(s)\mathbb{E}_{(x,a) \sim D, x_1' \sim \rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i\} |V(x_1', s, x, a)| \frac{\mathbb{E}_{x_2' \sim \rho_h}\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_2') = i \wedge g^\star(x_2') = s\}}{P_{s,i}}\right] \\
&= \frac{\rho_h(s)}{P_{s,i}}\mathbb{E}_{D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i = \hat{\phi}_h^{(\mathrm{B})}(x_2')\}\mathbf{1}\{g^\star(x_2') = s\} |V(x_1', x_2', x, a)|\right] \\
&\leq \frac{\rho_h(s)}{P_{s,i}}\mathbb{E}_{D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i = \hat{\phi}_h^{(\mathrm{B})}(x_2')\} |V(x_1', x_2', x, a)|\right] \leq \frac{8\rho_h(s)}{P_{s,i}}\sqrt{b_i \Delta_{reg}},
\end{aligned}$$

Putting things together we get for any policy π_1:

$$\left|\mathbb{P}[s \mid \pi_1] - \frac{\rho_h(s)}{b_i}\mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x'_1) = i \mid \pi_1]\right| \leq \frac{8N|\mathcal{A}|\rho_h(s)}{\alpha P_{s,i} b_i^{1/2}}\sqrt{\Delta_{reg}}.$$

Using the same bound for the second policy and combining the results we get the following inequality, which holds in the $1-\delta$ event of Corollary 6

$$\mathbb{P}[s \mid \pi_1] - \mathbb{P}[s \mid \pi_2] \leq \frac{\rho_h(s)}{b_i}\left(\mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i \mid \pi_1] - \mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i \mid \pi_2]\right) + \frac{16N|\mathcal{A}|\rho_h(s)}{\alpha P_{s,i} b_i^{1/2}}\sqrt{\Delta_{reg}}.$$

As this calculation applies for each i, we obtain the result. □

In the next lemma, we introduce our policy cover.

Lemma 12. *Assume that $\Psi_1, \ldots, \Psi_{h-1}$ are α policy covers for $\mathcal{S}_1, \ldots, \mathcal{S}_{h-1}$, each of size at most N. Let $\Psi_h := \{\hat{\pi}_{i,h} : i \in [N]\}$ be the policy cover learned at step h of* HOMER*. Then in the $\geq 1-(1+NH)\delta$ probability event that the N calls to* PSDP *succeed and Corollary 6 holds, we have that for any state $s \in \mathcal{S}_h$, there exists an index $i \in [N]$ such that*

$$\mathbb{P}[s \mid \hat{\pi}_{i,h}] \geq \eta(s) - \frac{N^2 h \Delta_{csc}}{\alpha} - \frac{16N^3|\mathcal{A}|^{3/2}}{\alpha^{3/2}}\sqrt{\Delta_{reg}/\eta(s)}$$

Proof. Let us condition on the success of Corollary 6, as well as the success of the N calls to PSDP. As the former fails with probability at most δ, and each call to PSDP fails with probability at most $H\delta$, the total failure probability here is $(1+NH)\delta$.

In this event, by Theorem 4, and the definition of the internal reward function R_i, we know that

$$\mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i|\hat{\pi}_{i,h}] \geq \max_{\pi \in \Pi_{\mathrm{NS}}} \mathbb{P}[\hat{\phi}_h^{(\mathrm{B})}(x') = i|\pi] - \frac{Nh\Delta_{csc}}{\alpha}.$$

Plugging this bound into Lemma 11, we get for any policy π

$$\mathbb{P}[s \mid \pi] \leq \mathbb{P}[s \mid \hat{\pi}_{i,h}] + \frac{Nh\rho_h(s)\Delta_{csc}}{\alpha b_i} + \frac{16N|\mathcal{A}|\rho_h(s)}{\alpha P_{s,i} b_i^{1/2}}\sqrt{\Delta_{reg}}.$$

This bound also holds for all $i \in [N]$. To optimize the bound, we should choose the index i that is maximally correlated with the state s. To do so, we choose $i(s) = \max_i P_{s,i}$. This index satisfies

$$b_{i(s)} = \sum_{s'} P_{s',i(s)} \geq P_{s,i(s)} = \max_i P_{s,i} \geq \frac{1}{N}\sum_{i=1}^{N} P_{s,i} = \frac{\rho_h(s)}{N}$$

Plugging in this bound, we see that for every s, there exists $i \in [N]$ such that

$$\eta(s) = \max_{\pi \in \Pi_{\mathrm{NS}}} \mathbb{P}[s \mid \pi] \leq \mathbb{P}[s \mid \hat{\pi}_{i,h}] + \frac{N^2 h \Delta_{csc}}{\alpha} + \frac{16N^{5/2}|\mathcal{A}|}{\alpha}\sqrt{\Delta_{reg}/\rho_h(s)}$$

We conclude the proof by introducing the lower bound on $\rho_h(s) \geq \frac{\alpha\eta(s)}{N|\mathcal{A}|}$ from Lemma 8 and re-arranging. □

D.3. Wrapping up the proof

Lemma 12 is the core technical result, which certifies that our learned policy cover at time h yields good coverage. We are basically done with the proof; all that remains is to complete the induction, set all of the parameters, and take a union bound.

Union bound. For each $h \in [H]$ we must invoke Corollary 6 once, and we invoke Theorem 4 N times. We also invoke Theorem 4 once more to learn the reward sensitive policy. Thus the total failure probability is $H(\delta_1 + NH\delta_2) + H\delta_3$ where δ_1 appears in Δ_{reg}, δ_2 appears in Δ_{csc} for the internal reward functions, and δ_3 appears in Δ_{csc} for the external reward functions. We therefore take $\delta_1 = \delta/(3H)$ and $\delta_2 = \frac{\delta}{3NH^2}$ and $\delta_3 = \frac{\delta}{3H}$, which gives us the settings

$$\Delta_{csc} = 4\sqrt{\frac{|\mathcal{A}|}{n_{\text{psdp}}} \ln\left(\frac{4NH^2|\Pi|}{\delta}\right)}, \qquad \Delta_{reg} = \frac{16\left(\ln(|\Phi_N|) + N^2|\mathcal{A}|\ln(n_{\text{reg}}) + \ln(6H/\delta)\right)}{n_{\text{reg}}},$$

for the inductive steps. With these choices, the total failure probability for the algorithm is δ.

The policy covers. Fix $h \in [H]$ and inductively assume that $\Psi_1, \dots, \Psi_{h-1}$ are $1/2$-policy covers for $\mathcal{S}_1, \dots, \mathcal{S}_{h-1}$. Then by Lemma 12, for each $s \in \mathcal{S}_h$ there exists $i \in [N]$ such that

$$\mathbb{P}[s \mid \hat{\pi}_{i,h}] \geq \eta(s) - 2N^2H\Delta_{csc} - 32\sqrt{2}N^3|\mathcal{A}|^{3/2}\sqrt{\Delta_{reg}/\eta(s)}.$$

We simply must set n_{psdp} and n_{reg} so that the right hand side here is at least $\eta(s)/2$. By inspection, sufficient conditions for both parameters are:

$$n_{\text{psdp}} \geq \frac{32^2N^4H^2|\mathcal{A}|}{\eta_{min}^2}\ln\left(\frac{4NH^2|\Pi|}{\delta}\right), \qquad 2\underbrace{n_{\text{reg}}}_{:=v} \geq \underbrace{\frac{512^2N^6|\mathcal{A}|^3}{\eta_{min}^3}}_{=:a}\left(\underbrace{N^2|\mathcal{A}|}_{=:c}\ln(n_{\text{reg}}) + \underbrace{\ln|\Phi_N| + \ln(6H/\delta)}_{=:b}\right).$$

To simplify the condition for n_{reg} we use the following transcendental inequality: For any $a > e$ and any b if $v \geq a\max\{c\ln(ac) + b, 0\}$ then $2v \geq ac\ln(v) + ab$. To see why, observe that

$$ac\ln(v) + ab = ac\ln(v/(ac)) + ac\ln(ac) + ab \leq v - ac + ac\ln(ac) + ab \leq 2v,$$

where the first inequality is simply that $\ln(x) \leq x - 1$ for $x > 0$, and the second inequality uses the lower bound on v. Using the highlighted definitions, a sufficient condition for n_{reg} is

$$n_{\text{reg}} \geq \frac{512^2N^6|\mathcal{A}|^3}{\eta_{min}^3}\left(N^2|\mathcal{A}|\ln\left(\frac{512^2N^8|\mathcal{A}|^4}{\eta_{min}^3}\right) + \ln|\Phi_N| + \ln(6H/\delta)\right).$$

Note that the algorithm sets these quantities in terms of a parameter η instead of η_{min}, which may not be known. As long as $\eta \leq \eta_{min}$ our settings of n_{psdp} and n_{reg} certify that Ψ_h is a $1/2$-policy cover for $\mathcal{S}_h$. Appealing to the induction, this establishes the policy cover guarantee.

The reward sensitive step. Equipped with the policy covers, a single call to PSDP with the external reward R and an application of Theorem 4 yield the PAC guarantee. We have already accounted for the failure probability, so we must simply set n_{eval}. Applying Theorem 4 with the definition of $\delta_3 = \delta/(3H)$, we get

$$n_{\text{eval}} \geq \frac{64N^2H^2|\mathcal{A}|}{\epsilon^2}\ln\left(\frac{3H|\Pi|}{\delta}\right).$$

Sample complexity. As we solve H supervised learning problem, make NH calls to PSDP with parameter n_{psdp}, and make 1 call to PSDP with parameter n_{eval}, the sample complexity, measured in trajectories, is

$$H \cdot n_{\text{reg}} + NH^2n_{\text{psdp}} + Hn_{\text{eval}} =$$
$$\widetilde{\mathcal{O}}\left(\frac{N^8|\mathcal{A}|^4H}{\eta_{min}^3} + \frac{N^6|\mathcal{A}|H}{\eta_{min}^3}\ln(|\Phi_N|/\delta) + \left(\frac{N^5H^4|\mathcal{A}|}{\eta_{min}^2} + \frac{N^2H^3|\mathcal{A}|}{\epsilon^2}\right)\ln(|\Pi|/\delta)\right).$$

Computational complexity. The running time is simply the time required to collect this many trajectories, plus the time required for all of the calls to the oracle. If T is the number of trajectories, the running time is

$$\mathcal{O}\left(HT + H\text{Time}_{\text{reg}}(n_{\text{reg}}) + NH^2\text{Time}_{\text{pol}}(n_{\text{psdp}}) + H\text{Time}_{\text{pol}}(n_{\text{eval}})\right).$$

D.4. Learning Forward Kinematic Inseparability

Our analysis of HOMER only considered the abstraction $\hat{\phi}_h^{(\mathrm{B})}$ which we showed learns the backward kinematic inseparability asymptotically and approximately in finite sample. This is sufficient for learning a policy cover and optimizing a given reward function However, for recovering the dynamics and visualization we also need forward kinematic inseparability abstraction. Here we argue that the $\hat{\phi}_{h-1}^{(\mathrm{F})}$ recovers forward kinematic inseparability asymptotically.

Lemma 13. *As $n \to \infty$, the learned abstraction $\hat{\phi}_{h-1}^{(\mathrm{F})}$ is a forward kinematic inseparability abstraction for $\mathcal{X}_{h-1}$.*

Proof. As we argued earlier, we can inductively assume Ψ_{h-1} is an α-policy cover for $\mathcal{S}_{h-1}$. This implies the distribution D in Corollary 6 has non-zero support over $\mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h$ and ρ_h has non-zero support over $\mathcal{X}_h$. Taking $n \to \infty$ in Corollary 6 then gives us $\hat{f}(x, a, x') = f^\star(x, a, x')$ for every $(x, a, x') \in \mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h$. Say $x_1, x_2 \in \mathcal{X}_{h-1}$ satisfy $\hat{\phi}_{h-1}^{(\mathrm{F})}(x_1) = \hat{\phi}_{h-1}^{(\mathrm{F})}(x_2)$. From the structure of $\hat{f}$ this implies

$$\forall a \in \mathcal{A}, x' \in \mathcal{X}_h, \qquad f^\star(x_1, a, x') = \hat{f}(x_1, a, x') = \hat{f}(x_2, a, x') = f^\star(x_2, a, x').$$

Using the form of $f^\star$ from Lemma 9 gives us:

$$\forall a \in \mathcal{A}, x' \in \mathcal{X}_h, \qquad \frac{T(x' \mid x_1, a)}{T(x' \mid x_1, a) + \rho_h(x')} = \frac{T(x' \mid x_2, a)}{T(x' \mid x_2, a) + \rho_h(x')} \Rightarrow T(x' \mid x_1, a) = T(x' \mid x_2, a)$$

Lastly, for $x' \notin \mathcal{X}_h$ and any $a \in \mathcal{A}$, we trivially have $T(x' \mid x_1, a) = T(x' \mid x_2, a) = 0$. Hence, $\hat{\phi}_{h-1}^{(\mathrm{F})}$ learns the forward kinematic inseparability abstraction asymptotically. □

E. Recovering State Abstraction from Non-Quantized Model Class via Clustering

In the main text we considered an oracle that recovers state abstractions $\{(\hat{\phi}_{h-1}^{(\mathrm{F})}, \hat{\phi}_h^{(\mathrm{B})})\}_{h=2}^H$ by training a model class with quantization. We showed that these state abstractions learn kinematic inseparability, and we showed that these optimization problems are empirically tractable in our experiments, despite nonconvexity. However, in general training a model class with quantization can be difficult. In this section, we show how to learn kinematic inseparability using a more standard square loss optimization primitive, without any quantization. We establish a oracle efficiency guarantee as well as a sample efficiency guarantee. However, the latter is worse than Theorem 1.

We consider a model class without quantization $\mathcal{F}_U : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \mapsto [0, 1]$ with U denoting *undiscretized*. We assume $\mathcal{F}_U$ is finite and our bounds scale with $\ln |\mathcal{F}_U|$. We do not make any assumption on this model class besides the realizability assumption $f_\rho \in \mathcal{F}_U$ for all $\rho \in \Delta(\mathcal{S}_h)$ with $\mathrm{supp}(\rho) = \mathcal{S}_h$, for all $h \in [H]$ (see Assumption 2).

We focus on recovering the backward kinematic inseparability $\hat{\phi}_h^{(\mathrm{B})}$ from training with model class $\mathcal{F}_U$. The forward KI abstraction $\hat{\phi}_{h-1}^{(\mathrm{F})}$ can be recovered similarly but we omit this treatment for brevity as HOMER only relies on $\hat{\phi}_h^{(\mathrm{B})}$ for the sample complexity guarantees.

Consider the h^{th} iteration of HOMER (Algorithm 1, line 2-line 15). Let D be a dataset of real and imposter transitions in the h^{th} iteration (Algorithm 1, line 4-line 10). Let $\hat{f}$ be the empirical risk minimizer (ERM) given by:

$$\hat{f} = \texttt{REG}(\mathcal{F}_U, D) = \operatorname*{argmin}_{f \in \mathcal{F}_U} \sum_{([x,a,x'],y) \in D} \left(f(x, a, x') - y\right)^2. \tag{3}$$

Our computational assumption will be that $\texttt{REG}(\mathcal{F}_U, D)$ can be implemented efficiently. This is a standard squared loss minimization problem so we expect it to be easier than $\texttt{REG}(\mathcal{F}_N, D)$ as $\mathcal{F}_U$ is not quantized.

As we do not assume a bottleneck structure in $\mathcal{F}_U$, therefore, we need to find a way to recover $\hat{\phi}_h^{(\mathrm{B})}$ from $\hat{f}$. We achieve this using the algorithm ClusteredBKI (see Algorithm 3). ClusteredBKI takes as input the dataset D, the model class $\mathcal{F}_U$, the policy cover Ψ_{h-1} for the previous time step, the estimate η of η_{min} and failure probability δ. The algorithm returns the abstraction $\hat{\phi}_h^{(\mathrm{B})}$ that we will show learns the backward kinematic inseparability, analogously to our proof using the quantization oracle. This oracle can be easily used with HOMER by replacing a call to $\texttt{REG}(\mathcal{F}_N, D)$ (Algorithm 1, line 11) with a call to $\texttt{ClusteredBKI}(\mathcal{F}_U, D, \Psi_{h-1}, \eta, \delta)$. The returned $\hat{\phi}_h^{(\mathrm{B})}$ can be used as before.

Algorithm 3 $\texttt{ClusteredBKI}(\mathcal{F}_U, D, \Psi_{h-1}, \eta, \delta)$ learns a state abstraction function using noise contrastive estimation with model class $\mathcal{F}_U$ without quantization, a dataset of real and imposter transitions D, a set of exploration policies for previous time step Ψ_{h-1}, an estimate η of η_{min}, and failure probability δ.

1: Set $N = |\Psi_{h-1}|$, $n = |D|$, $m = \frac{2N|\mathcal{A}|}{\alpha\eta} \ln\left(\frac{eN^2|\mathcal{A}|}{\delta}\right)$ and $\tau = \frac{8m^3}{\delta^{3/2}}\left(\frac{32}{n}\ln\left(\frac{2|\mathcal{F}_U|}{\delta}\right)\right)^{\frac{1}{4}}$
2: $\hat{f} = \texttt{REG}(\mathcal{F}_U, D)$ // Perform regression on $\mathcal{D}$ with model class $\mathcal{F}_U$
3: Let $\mathcal{U} = (x'^{(1)}, \cdots, x'^{(m)})$ where $(x^{(i)}, a^{(i)}, x'^{(i)}) \sim \texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$
4: Let $\mathcal{V} = ((x^{(1)}, a^{(1)}), \cdots, (x^{(m)}, a^{(m)}))$ where $(x^{(i)}, a^{(i)}, x'^{(i)}) \sim \texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$
5: Define a feature function $\hat{\xi} : \mathcal{X} \to \mathbb{R}^m$ as
$$\forall x' \in \mathcal{X}, \qquad \hat{\xi}(x') = (\hat{f}(x^{(1)}, a^{(1)}, x'), \cdots, \hat{f}(x^{(m)}, a^{(m)}, x'))$$
6: $(c_i)_{i=1}^k = \texttt{GreedyClustering}(\{\hat{\xi}(x') \mid x' \in \mathcal{U}\}, \|.\|_1, \tau)$ // Cluster features generated by applying $\hat{\xi}$ to $\mathcal{U}$
7: Define a state abstraction function $\hat{\phi}_h^{(\mathrm{B})} : \mathcal{X} \to \mathbb{N}$ as
$$\forall x' \in \mathcal{X}, \qquad \hat{\phi}_h^{(\mathrm{B})}(x') = \arg\min_{i \in [k]} \|\hat{\xi}(x') - c_i\|_1$$
8: **return** $\hat{\phi}_h^{(\mathrm{B})}$

Algorithm 4 $\texttt{GreedyClustering}(\mathcal{Z}, \|.\|, \tau)$ Cluster vectors $\mathcal{Z}$ using distance function $\|.\|$ and threshold τ

1: Set $\mathcal{C} = \emptyset$ // Set of cluster centers
2: **while** $\mathcal{Z} \neq \emptyset$ **do**
3: Pick any z from $\mathcal{Z}$
4: $\mathcal{C} \leftarrow \mathcal{C} \cup \{z\}$
5: Define $\mathcal{B}_\tau(z) = \{z' \in \mathcal{Z} \mid \|z - z'\| < \tau\}$ // Set of points remaining in $\mathcal{Z}$ that are covered by z
6: $\mathcal{Z} = \mathcal{Z} - \mathcal{B}_z(\tau)$
7: **return** $\mathcal{C}$

$\texttt{ClusteredBKI}$ first computes the ERM solution in Equation 3 (Algorithm 3, line 2). We then sample a set of observations $\mathcal{U}$ and observation-action tuples $\mathcal{V}$ using the sampling procedure $\texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$. Recall that $(x, a, x') \sim \texttt{Unf}(\Psi_{h-1}) \circ \texttt{Unf}(\mathcal{A})$ is generated by uniformly sampling a policy in Ψ_{h-1} and roll-in with it for $h-1$ steps to observe x, and taking action a uniformly in x to observe x'. The set $\mathcal{U}$ contains x' and $\mathcal{V}$ contains (x, a) sampled this way (Algorithm 3, line 3-line 4). Both these sets contain i.i.d. entries and are independent of each other.

The key step to recovering $\hat{\phi}_h^{(\mathrm{B})}$ is to perform clustering on a set of features where each cluster would represent an abstract state. We first use $\hat{f}$ and the set $\mathcal{V}$ to define a feature function $\hat{\xi} : \mathcal{X} \to \mathbb{R}^m$ (Algorithm 3, line 5). For any observation $x' \in \mathcal{X}$, the feature $\hat{\xi}(x')$ is a vector of values $\hat{f}(x^{(i)}, a^{(i)}, x')$ where $(x^{(i)}, a^{(i)}) \in \mathcal{V}$. We do clustering on the set of features $\{\hat{\xi}(x') \mid x' \in \mathcal{U}\}$ generated by observations in $\mathcal{U}$ using the subroutine $\texttt{GreedyClustering}$ (Algorithm 4). Intuitively, if $\mathcal{V}$ has good coverage over states and actions at time step $h-1$, and if $\hat{f}$ is trained sufficiently well then we can hope that observations which are backward KI will have features which are close to each other. Similarly, for observations which are not backward KI, we can hope the features will differ for at least some $(x^{(i)}, a^{(i)}) \in \mathcal{V}$.

The clustering subroutine $\texttt{GreedyClustering}$ takes as input a set of features $\mathcal{Z}$, a distance function $\|.\|$ and a real number $\tau > 0$ denoting the size of clusters. We use L_1 distance as our choice of distance function (Algorithm 3, line 6). Further, we set τ based on the size of dataset D which implicitly controls the generalization error of $\hat{f}$ (Algorithm 3, line 1). The clustering subroutine repeatedly picks a feature z in $\mathcal{Z}$ and marks it as a cluster center by adding it to a set $\mathcal{C}$ (Algorithm 4, line 3-line 4). It then removes every feature remaining in $\mathcal{Z}$ within τ distance of z, which includes z, from $\mathcal{Z}$ (Algorithm 4, line 6). The removed features can be considered as assigned to the cluster center z. This is repeated until $\mathcal{Z}$ is empty which happens in at most $|\mathcal{Z}|$ iterations (Algorithm 4, line 2-line 6). The cluster centers $\mathcal{C}$ are returned as output.

The learned state abstraction $\hat{\phi}_h^{(\mathrm{B})} : \mathcal{X} \to$ maps a given observation x' to the identity of the cluster center closest to $\hat{\xi}(x')$ (Algorithm 3, line 7). `ClusteredBKI` returns $\hat{\phi}_h^{(\mathrm{B})}$ as output.

E.1. Sample Complexity Analysis

We inductively focus on the task of learning $\hat{\phi}_h^{(\mathrm{B})}$ assuming we have an α-policy cover Ψ_{h-1} for previous time step of size N. This is similar to our analysis in Appendix D. We first analyze the performance of $\hat{f}$. Recall that we learn $\hat{f}$ using a dataset of real and imposter transitions sampled from $D \in \Delta(\mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h \times \{0,1\})$. Further, recall that $\rho_h(x') := \mathbb{P}_{\pi \circ \mathtt{Unf}:\pi \sim \mathtt{Unf}(\Psi_{h-1})}[x']$ and $\mu_{h-1}(x) := \mathbb{P}_{\pi \sim \mathtt{Unf}(\Psi_{h-1})}[x]$. Let $D(x,a,x')$ is the marginal distribution over real and imposter transitions then:

$$D(x,a,x') = \frac{\mu_{h-1}(x)}{2|\mathcal{A}|} \cdot \{T(x' \mid x,a) + \rho_h(x')\}.$$

First, the standard supervised learning guarantees apply as stated in Theorem 8:

Theorem 8. *Fix $\delta \in (0,1)$. Let D be a dataset of n i.i.d. real and imposter transitions and let $\hat{f}$ be the ERM solution (Equation 3). Then the following holds with probability at least $1-\delta$:*

$$\mathbb{E}_{x \sim \mu_{h-1}, a \sim \mathtt{Unf}, x' \sim \rho_h}\left[|\hat{f}(x,a,x') - f^\star(x,a,x')|\right] \le \Delta_{cerr}(n,\delta), \qquad \text{where } \Delta_{cerr}(n,\delta) = \sqrt{\frac{32}{n}\ln\left(\frac{2|\mathcal{F}_U|}{\delta}\right)}. \quad (4)$$

Proof. We use Proposition 11 for finite classes to first get the following with probability at least $1-\delta$:

$$\mathbb{E}_{x,a,x' \sim D}\left[\left(\hat{f}(x,a,x') - f^\star(x,a,x')\right)^2\right] \le \frac{8}{n}\ln\left(\frac{2|\mathcal{F}_U|}{\delta}\right).$$

Here we use a trivial observation that $\mathcal{N}(\mathcal{F}_U,\epsilon) \le |\mathcal{F}_U|$ for any $\epsilon \ge 0$. Next we use Jensen's inequality to get:

$$\mathbb{E}_{x,a,x' \sim D}\left[|\hat{f}(x,a,x') - f^\star(x,a,x')|\right] \le \sqrt{\mathbb{E}_{x,a,x' \sim D}\left[\left(\hat{f}(x,a,x') - f^\star(x,a,x')\right)^2\right]} \le \sqrt{\frac{8}{n}\ln\left(\frac{2|\mathcal{F}_U|}{\delta}\right)}.$$

Lastly, using $D(x,a,x') \ge \frac{\mu_{h-1}(x)\rho_h(x')}{2|\mathcal{A}|}$ we get the following with at least $1-\delta$ probability:

$$\mathbb{E}_{x \sim \mu_{h-1}, a \sim \mathtt{Unf}, x' \sim \rho_h}\left[|\hat{f}(x,a,x') - f^\star(x,a,x')|\right] \le \sqrt{\frac{32}{n}\ln\left(\frac{2|\mathcal{F}_U|}{\delta}\right)}.$$

□

For a given $\mathcal{V}$, we define the "correct" feature function $\xi : \mathcal{X} \to \mathbb{R}^m$ as:

$$\forall x' \in \mathcal{X}, \qquad \xi(x') = \left(f^\star(x^{(1)},a^{(1)},x'), \cdots, f^\star(x^{(m)},a^{(m)},x')\right).$$

Our feature function $\hat{\xi}$ is trying to approximate ξ. If we had access to ξ then clustering the features $\{\xi(x') \mid x' \in \mathcal{U}\}$ with $\tau = 0$ would give us a cluster center for every backward KI state at timestep h, provided $\mathcal{U}$ and $\mathcal{V}$ have good coverage. This holds since ξ uses $f^\star$ which only depends upon the KI state identity and not the actual observation. Our analysis will prove that using $\hat{\xi}$ instead of ξ also works with high probability.

E.1.1. Error Bounds for sampled $\mathcal{U}$ and $\mathcal{V}$.

We want to bound the error induced due to imperfect $\hat{f}$ for sampled $\mathcal{V}$ and $\mathcal{U}$.

Lemma 14 (Joint Conditional Error)**.** *Let $\mathcal{V} = ((x^{(1)},a^{(1)}),\cdots,(x^{(m)},a^{(m)}))$. Then under success probability of Theorem 8 the following holds for any $e > 0$:*

$$\forall k \in [m], \qquad \mathbb{E}_{x' \sim \rho_h}\left[\left|\hat{f}(x^{(k)},a^{(k)},x') - f^\star(x^{(k)},a^{(k)},x')\right|\right] < e, \quad (5)$$

with probability at least $1 - \frac{m\Delta_{cerr}(n,\delta)}{e}$.

Proof. For any $k \in [m]$, let $W(x^{(k)}, a^{(k)}) = \mathbb{E}_{x' \sim \rho_h}\left[\left|\hat{f}(x^{(k)}, a^{(k)}, x') - f^\star(x^{(k)}, a^{(k)}, x')\right|\right]$. Then from Equation 4 we have $\mathbb{E}_{x \sim \mu_{h-1}, a \sim \texttt{Unf}}[W(x, a)] \leq \Delta_{cerr}(n, \delta)$. We want to bound the following quantity:

$$\begin{aligned}
\mathbb{P}_{\mathcal{V}}\left(\cap_{k=1}^m W(x^{(k)}, a^{(k)}) < e\right) &\geq 1 - \mathbb{P}_{\mathcal{V}}\left(\cup_{k=1}^m W(x^{(k)}, a^{(k)}) \geq e\right) \\
&\geq 1 - \sum_{k=1}^m \mathbb{P}_{\mathcal{V}}\left(W(x^{(k)}, a^{(k)}) \geq e\right), && \text{(using union bound)} \\
&\geq 1 - \frac{1}{e}\sum_{k=1}^m \mathbb{E}_{x,a \sim \mu_h}\left[W(x, a)\right], && \text{(using Markov's inequality)} \\
&\geq 1 - \frac{m\Delta_{cerr}(n, \delta)}{e}. && \text{(using Equation 4)} \quad \square
\end{aligned}$$

We use Lemma 14 to bound the expected error in $\hat{\xi}$.

Lemma 15 (Expected Error in $\hat{\xi}$)**.** *Fix $e > 0$ then under success probability of Theorem 8 the following holds with probability of at least $1 - \frac{m\Delta_{cerr}(n,\delta)}{e}$:*

$$\mathbb{E}_{x' \sim \rho_h}\left[\|\xi(x') - \hat{\xi}(x')\|_1\right] \leq 2\sqrt{m^3 e}.$$

Proof. Let $a > 0$ be any number then:

$$\begin{aligned}
\mathbb{E}_{x'_1 \sim \rho_h}\left[\|\xi(x'_1) - \hat{\xi}(x'_1)\|_1\right] &\leq ma + m\mathbb{P}_{x'}(\|\xi(x') - \hat{\xi}(x')\|_1 > ma), && \text{(using } \|\xi(x') - \hat{\xi}(x')\|_1 \leq m) \\
&\leq ma + m\mathbb{P}_{x'}\left(\cup_{k=1}^m \left|\hat{f}(x^{(k)}, a^{(k)}, x') - f^\star(x^{(k)}, a^{(k)}, x')\right| > a\right) \\
&\leq ma + m\sum_{k=1}^m \mathbb{P}_{x'}\left(\left|\hat{f}(x^{(k)}, a^{(k)}, x') - f^\star(x^{(k)}, a^{(k)}, x')\right| > a\right), && \text{(using union bound)} \\
&\leq ma + \frac{m}{a}\sum_{k=1}^m \mathbb{E}_{x' \sim \rho_h}\left[\left|\hat{f}(x^{(k)}, a^{(k)}, x') - f^\star(x^{(k)}, a^{(k)}, x')\right|\right], && \text{(using Markov's inequality)} \\
&\leq ma + \frac{m^2 e}{a}, && \text{(using Lemma 14).}
\end{aligned}$$

The value of a that gives the tightest bound is $\sqrt{me}$ which gives an upper bound of $2\sqrt{m^3 e}$. The only probabilistic statement we use is Lemma 14 which holds with probability of at least $1 - \frac{m\Delta_{cerr}(n,\delta)}{e}$. $\square$

Lemma 16. *(Pointwise Error in $\hat{\xi}$) Let $\mathcal{U} = \{x'^{(1)}, \cdots, x'^{(m)}\}$. Fix $u > 0$. Then under success probability of Theorem 8 the following holds with probability at least $1 - \frac{m\Delta_{cerr}(n,\delta)}{e} - \frac{2m\sqrt{m^3 e}}{u}$:*

$$\forall i \in [m], \qquad \|\xi(x'^{(i)}) - \hat{\xi}(x'^{(i)})\|_1 < u.$$

Proof. Let X_i be the event that $\|\xi(x'^{(i)}) - \hat{\xi}(x'^{(i)})\|_1 < u$. Recall $x^{(1)}, \cdots, x^{(m)}$ are independent samples from ρ_h. We want to bound:

$$\begin{aligned}
\mathbb{P}(\cap_{i=1}^m X_i) = 1 - \mathbb{P}(\cup_{i=1}^m \overline{X_i}) &\geq 1 - \sum_{i=1}^m \mathbb{P}(\overline{X_i}), && \text{(using union bound)} \\
&= 1 - \sum_{i=1}^m \mathbb{P}(\|\xi(x'^{(i)}) - \hat{\xi}(x'^{(i)})\|_1 \geq u) \\
&\geq 1 - \sum_{i=1}^m \frac{1}{u}\mathbb{E}_{x' \sim \rho_h}\left[\|\xi(x') - \hat{\xi}(x')\|_1\right], && \text{(using Markov's inequality)} \\
&\geq 1 - \frac{2m\sqrt{m^3 e}}{u}. && \text{(using Lemma 15)}
\end{aligned}$$

The failure probability due to Lemma 15 is $\frac{m\Delta_{cerr}(n,\delta)}{e}$ hence the total failure probability is $\frac{m\Delta_{cerr}(n,\delta)}{e} + \frac{2m\sqrt{m^3 e}}{u}$. $\square$

E.1.2. COVERAGE LEMMAS.

Let there be $\mathrm{N}_{\mathrm{KD}}^{(t)}$ many kinematically inseparable abstract states at time step $t \in [H]$. We know that observations emitted by a state are kinematically inseparable so $\mathrm{N}_{\mathrm{KD}}^{(t)} \leq |\mathcal{S}_t|$. Also, trivially we have $\mathrm{N}_{\mathrm{KD}}^{(t)} \leq \mathrm{N}_{\mathrm{KD}}$. We denote KI states by index in the set $\bar{\mathcal{S}}_t = [\mathrm{N}_{\mathrm{KD}}^{(t)}]$ and let $\phi_t^\star : \mathcal{X} \to \bar{\mathcal{S}}_t$ be the KI abstraction function.

We start by showing that $\mathcal{V}$ has good coverage over $\bar{\mathcal{S}}_{h-1} \times \mathcal{A}$ with high probability.

Lemma 17 (Coverage Lemma for $\mathcal{V}$)**.** *Fix $d \in \mathbb{N}$ and $\delta_c \in (0,1)$. Let $\mathcal{V} = ((x^{(1)}, a^{(1)}), \cdots, (x^{(m)}, a^{(m)}))$. Then with at least $1 - \delta_c$ probability, for each $j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}$ we have $|\{(x^{(i)}, a^{(i)}) \in \mathcal{V} \mid \phi_{h-1}^\star(x^{(i)}) = j \wedge a^{(i)} = a\}| \geq d$ if*

$$m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \left\{ d + \ln\left(\frac{N_{KD}^{(h-1)}|\mathcal{A}|}{\delta_c} \right) \right\}. \tag{6}$$

Proof. The proof is a standard Chernoff bound argument. Let $X_{j,a}$ be the event $|\{(x^{(i)}, a^{(i)}) \in \mathcal{V} \mid \phi_{h-1}^\star(x^{(i)}) = j \wedge a^{(i)} = a\}| \geq d$. Let $Z_{j,a}^{(i)}$ be a 0-1 random variable that is 1 if $\phi_{h-1}^\star(x^{(i)}) = j$ and $a^{(i)} = a$ and 0 otherwise. Our aim is to bound:

$$\mathbb{P}(\cap_{j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}} X_{j,a}) = 1 - \mathbb{P}(\cup_{j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}} \overline{X_{j,a}}) \geq 1 - \sum_{j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}} \mathbb{P}(\overline{X_{j,a}}) = 1 - \sum_{j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}} \mathbb{P}\left(\sum_{i=1}^{m} Z_{j,a}^{(i)} < d \right).$$

For any $j \in \bar{\mathcal{S}}_{h-1}$ and $a \in \mathcal{A}$, $\left\{ Z_{j,a}^{(1)}, \cdots, Z_{j,a}^{(m)} \right\}$ are 0-1 i.i.d. random variables. Define $\theta_{j,a} := \mathbb{E}_{\mathcal{V}}[Z_{j,a}^{(i)}]$. We derive a lower bound on the value of $\theta_{j,a}$:

$$\theta_{j,a} = \mathbb{P}_{\tilde{x},\tilde{a}}(\phi_{h-1}^\star(\tilde{x}) = j \wedge \tilde{a} = a) = \mathbb{P}_{\tilde{x}}(\phi_{h-1}^\star(\tilde{x}) = j)\mathbb{P}_{\tilde{a}}(\tilde{a} = a) \geq \frac{\alpha\eta_{min}}{N|\mathcal{A}|},$$

where we use the fact that Ψ_{h-1} is an α-policy cover of $\mathcal{S}_{h-1}$, any KI abstract state j contains at least one real state, $|\Psi_{h-1}| = N$ and actions are taken uniformly. Let $\omega = 1 - \frac{d}{m\theta_{j,a}}$. We will assume $\omega > 0$ i.e., $m \geq \frac{d}{\theta_{j,a}} \geq \frac{dN|\mathcal{A}|}{\alpha\eta_{min}}$. This can be ensured by choosing a sufficiently large value of m. Then using multiplicative Chernoff's bound we have:

$$\mathbb{P}\left(\sum_{i=1}^{m} Z_{j,a}^{(i)} < d \right) = \mathbb{P}\left(\sum_{i=1}^{m} Z_{j,a}^{(i)} < (1-\omega) m\theta_{j,a} \right) \leq \exp\left\{ -\frac{m\theta_{j,a}\omega^2}{2} \right\}.$$

We can bound the upper bound using lower bound on $\theta_{j,a}$.

$$\exp\left\{ -\frac{m\theta_{j,a}\omega^2}{2} \right\} = \exp\left\{ -\frac{m\theta_{j,a}}{2} - \frac{d^2}{2m\theta_{j,a}} + d \right\} \leq \exp\left\{ -\frac{m\theta_{j,a}}{2} + d \right\} \leq \exp\left\{ d - \frac{\alpha m\eta_{min}}{2N|\mathcal{A}|} \right\}.$$

Plugging this bound we get:

$$\mathbb{P}(\cap_{j \in \bar{\mathcal{S}}_{h-1}, a \in \mathcal{A}} X_{j,a}) \geq 1 - \mathrm{N}_{\mathrm{KD}}^{(h-1)}|\mathcal{A}| \exp\left\{ d - \frac{\alpha m\eta_{min}}{2N|\mathcal{A}|} \right\}.$$

As we want the failure probability to be at most δ_c therefore, we set:

$$\mathrm{N}_{\mathrm{KD}}^{(h-1)}|\mathcal{A}| \exp\left\{ d - \frac{\alpha m\eta_{min}}{2N|\mathcal{A}|} \right\} \leq \delta_c \Rightarrow m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \left\{ d + \ln\left(\frac{\mathrm{N}_{\mathrm{KD}}^{(h-1)}|\mathcal{A}|}{\delta_c} \right) \right\}.$$

This bound also satisfies $m \geq \frac{d}{\theta_{j,a}} \geq \frac{dN|\mathcal{A}|}{\alpha\eta_{min}}$ needed for application of Chernoff's bound. □

Similarly, we can prove a coverage result for $\mathcal{U}$.

Lemma 18 (Coverage Lemma for $\mathcal{U}$). *Fix $d \in \mathbb{N}$ and $\delta_c \in (0,1)$. Let $\mathcal{U} = \{x'^{(1)}, \cdots, x'^{(m)}\}$. Then with at least $1-\delta_c$ probability, for each $j \in \bar{\mathcal{S}}_h$ we have $|\{x'^{(i)} \in \mathcal{U} \mid \phi_h^\star(x'^{(i)}) = j\}| \geq d$ if*

$$m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \left\{ d + \ln\left(\frac{N_{KD}^{(h)}}{\delta_c}\right)\right\}. \tag{7}$$

Proof. Proof is similar to Lemma 17. For any $j \in \bar{\mathcal{S}}_h$ let X_j be the event $|\{x'^{(i)} \in \mathcal{U} | \phi_h^\star(x'^{(i)}) = j\}| \geq d$ and $Z_j^{(i)}$ is a 0-1 random variable which is 1 iff $\phi_h^\star(x'^{(i)}) = j$. Then similar to Lemma 17 we have

$$\mathbb{P}(\cap_{j\in\bar{\mathcal{S}}_h} X_j) = 1 - \mathbb{P}(\cup_{j\in\bar{\mathcal{S}}_h} \overline{X_j}) \geq 1 - \sum_{j\in\bar{\mathcal{S}}_h} \mathbb{P}(\overline{X_j}) = 1 - \sum_{j\in\bar{\mathcal{S}}_h} \mathbb{P}\left(\sum_{i=1}^{m} Z_j^{(i)} < d\right).$$

For any $j \in \bar{\mathcal{S}}_h$, $\{Z_j^{(1)}, \cdots, Z_j^{(m)}\}$ are 0-1 random variables. Define $\beta_j := \mathbb{E}_{\mathcal{U}}[Z_j^{(i)}]$. We derive a lower bound on β_j below:

$$\beta_j = \mathbb{E}_{\mathcal{U}}[Z_j^{(i)}] = \mathbb{E}_{x'\sim\rho_h}[x' = j] \geq \frac{\alpha\eta_{min}}{N|\mathcal{A}|},$$

where we use the fact that abstract state j contains at least one state (say s_j) and the lower bound on $\rho_h(s_j)$ from Lemma 8.

This lower bound is the same for $\theta_{j,a}$ in Lemma 17 therefore, we can borrow the calculations we did there to get:

$$\mathbb{P}\left(\sum_{i=1}^{m} Z_j^{(i)} < d\right) \leq \exp\left\{d - \frac{\alpha m\eta_{min}}{2N|\mathcal{A}|}\right\} \Rightarrow \mathbb{P}\left(\cap_{j\in\bar{\mathcal{S}}_h} X_j\right) \geq 1 - \mathrm{N}_{\mathrm{KD}}^{(h)} \exp\left\{d - \frac{\alpha m\eta_{min}}{2N|\mathcal{A}|}\right\},$$

setting the failure probability to be at most δ_c gives us the derived result. □

Combining Lemma 18 and Lemma 17 we get a lower bound on m of:

$$m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \left\{ d + \max\left\{ \ln\left(\frac{\mathrm{N}_{\mathrm{KD}}^{(h-1)}|\mathcal{A}|}{\delta_c}\right), \ln\left(\frac{\mathrm{N}_{\mathrm{KD}}^{(h)}}{\delta_c}\right)\right\}\right\},$$

which can be weakend into a simpler form below using $\max\{\ln(a), \ln(b)\} \leq \ln(a) + \ln(b) = \ln(ab)$ for $a, b \geq 1$ and $N \geq \mathrm{N_{KD}} \geq \max\{\mathrm{N}_{\mathrm{KD}}^{(h-1)}, \mathrm{N}_{\mathrm{KD}}^{(h)}\}$. In summary, the bound is

$$m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \ln\left(\frac{e^d N^2|\mathcal{A}|}{\delta_c}\right). \tag{8}$$

E.1.3. Other Lemmas.

We now want to bound the difference between $\|\xi(x'_1) - \xi(x'_2)\|_1$ for two observations that are mapped to the same abstract state in terms of difference between their $f^\star$ values. We achieve this with the next result. Recall the coupling distribution $D_{\text{coup}} \in \Delta(\mathcal{X}_{h-1} \times \mathcal{A} \times \mathcal{X}_h \times \mathcal{X}_h)$ introduced in Appendix D. For any given (x, a, x'_1, x'_2), we have $D_{\text{coup}}(x, a, x'_1, x'_2) = \mu_{h-1}(x)\frac{1}{|\mathcal{A}|}\rho_h(x'_1)\rho_h(x'_2)$. We define the distribution $D \in \Delta(\mathcal{X}_{h-1} \times \mathcal{A})$ as $D(x, a) = \mu_{h-1}(x)\frac{1}{|\mathcal{A}|}$. We can lift D to states or abstract states in the natural way. It is straightforward to see that $D_{\text{coup}}(x, a, x'_1, x'_2) = D(x, a)\rho_h(x'_1)\rho_h(x'_2)$.

Lemma 19 (Difference in $f^\star$). *Fix $\delta_c \in (0,1)$ and $d \in \mathbb{N}$. Let $\mathcal{V}$ be the sampled set and $\hat{\phi}_h^{(\mathrm{B})}$ be the learned state abstraction. Assume $|\mathcal{V}| = m$ satisfies lower bound in Equation 8. Then the following holds:*

$$\mathbb{E}_{x,a,x'_1,x'_2\sim D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\,|f^\star(x, a, x'_1) - f^\star(x, a, x'_2)|\right] \leq$$
$$\frac{1}{d}\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\xi(x'_1) - \xi(x'_2)\|_1\right],$$

with probability of at least $1-\delta_c$ over draws of $\mathcal{V}$.

Proof. Let $c(j,a) = |\{(x^{(k)}, a^{(k)}) \in \mathcal{V} \mid \phi^{\star}_{h-1}(x^{(k)}) = j, a^{(k)} = a\}|$. Recall that ξ and $f^{\star}$ only depend upon the KI abstract state identity and not the observation. Therefore,

$$
\begin{aligned}
&\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}\|\xi(x'_1) - \xi(x'_2)\|_1\right] \\
&= \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}\sum_{k=1}^{m}|f^{\star}(x^{(k)},a^{(k)},x'_1) - f^{\star}(x^{(k)},a^{(k)},x'_2)|\right] \\
&= \sum_{k=1}^{m}\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}|f^{\star}(x^{(k)},a^{(k)},x'_1) - f^{\star}(x^{(k)},a^{(k)},x'_2)|\right] \\
&= \sum_{j\in\bar{\mathcal{S}}_{h-1}}\sum_{a\in\mathcal{A}} c(j,a)\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}|f^{\star}(j,a,x'_1) - f^{\star}(j,a,x'_2)|\right] \\
&= \sum_{j\in\bar{\mathcal{S}}_{h-1}}\sum_{a\in\mathcal{A}} \frac{c(j,a)}{D(j,a)}D(j,a)\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}|f^{\star}(j,a,x'_1) - f^{\star}(j,a,x'_2)|\right] \\
&\geq \inf_{j',a'}\frac{c(j',a')}{D(j',a')}\mathbb{E}_{j,a,x'_1,x'_2\sim D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}|f^{\star}(j,a,x'_1) - f^{\star}(j,a,x'_2)|\right] \\
&\geq d\mathbb{E}_{x,a,x'_1,x'_2\sim D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x'_1) = \hat{\phi}^{(\mathrm{B})}_h(x'_2)\}|f^{\star}(x,a,x'_1) - f^{\star}(x,a,x'_2)|\right]. \\
&\qquad \text{(using } c(j',a') \geq d \text{ from Lemma 17, } D_{\text{coup}}(j',a') \leq 1 \text{ and putting } x \text{ back instead of } j)
\end{aligned}
$$

The proof is completed by noting that Lemma 17 holds with probability at least $1-\delta_c$. □

We now prove the result for the clustering oracle that will be useful later. Note that the clustering subroutine (Algorithm 4) provides the following guarantee:

$$\text{Clustering Guarantee:} \quad \forall x' \in \mathcal{U},\ \exists c_i,\ \text{such that } \|\hat{\xi}(x') - c_i\|_1 < \tau \tag{9}$$

Lemma 20 (Clustering Performance)**.** *Fix $e, u > 0$ and learned abstract state $j \in \mathbb{N}$. Then*

$$\mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\|\hat{\xi}(x') - c_j\|_1\right] \leq 2\sqrt{m^3 e} + u + b_j\tau, \tag{10}$$

with probability at least $1 - \delta_c - \frac{m\Delta_{cerr}(n,\delta)}{e} - \frac{2\sqrt{m^5 e}}{u}$.

Proof. From Lemma 18 there exists at least d observations for any KI state in $\bar{\mathcal{S}}_h$ with probability at least $1-\delta_c$ for any $\delta_c \in (0,1)$ and $d \geq 1$, provided m satisfies bound in Equation 8. We can therefore, assign an observation $x' \in \mathcal{X}_h$ to an observation $\kappa(x') \in \mathcal{U}$ that has the same KI state. Observe that $\hat{\xi}(\kappa(x'))$ would be one of the vectors that is given as input to the clustering algorithm (Algorithm 3, line 6).

$$
\begin{aligned}
&\mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\|\hat{\xi}(x') - c_j\|_1\right] \\
&= \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\min_{i\in[k]}\|\hat{\xi}(x') - c_i\|_1\right], \qquad \text{(using the definition of } \hat{\phi}(x') = \arg\min_{i\in[k]}\|\hat{\xi}(x') - c_i\|_1) \\
&= \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\min_{i\in[k]}\left\{\|\hat{\xi}(x') - c_i\|_1 - \|\hat{\xi}(\kappa(x')) - c_i\|_1 + \|\hat{\xi}(\kappa(x')) - c_i\|_1\right\}\right] \\
&\leq \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\min_{i\in[k]}\left\{\|\hat{\xi}(x') - \hat{\xi}(\kappa(x'))\|_1 + \|\hat{\xi}(\kappa(x')) - c_i\|_1\right\}\right], \qquad \text{(using triangle inequality)} \\
&= \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\|\hat{\xi}(x') - \hat{\xi}(\kappa(x'))\|_1\right] + \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\min_{i\in[k]}\|\hat{\xi}(\kappa(x')) - c_i\|_1\right] \\
&\leq \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\|\hat{\xi}(x') - \hat{\xi}(\kappa(x'))\|_1\right] + \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\left\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\right\}\tau\right], \\
&\quad \text{(using clustering guarantee in Equation 9)} \\
&\leq \mathbb{E}_{x'\sim\rho_h}\left[\|\hat{\xi}(x') - \hat{\xi}(\kappa(x'))\|_1\right] + b_j\tau, \\
&\quad \text{(using the definition of } b_j = \mathbb{E}_{x'\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\}\right] \text{ and } \mathbf{1}\{\hat{\phi}^{(\mathrm{B})}_h(x') = j\} \leq 1).
\end{aligned}
$$

We can bound the first term as follows:

$$\mathbb{E}_{x'\sim\rho_h}\left[\|\hat{\xi}(x')-\hat{\xi}(\kappa(x'))\|_1\right] \leq \underbrace{\mathbb{E}_{x'\sim\rho_h}\left[\|\hat{\xi}(x')-\xi(x')\|_1\right]}_{\text{Term 1}} + \underbrace{\mathbb{E}_{x'\sim\rho_h}\left[\|\xi(x')-\xi(\kappa(x'))\|_1\right]}_{\text{Term 2}} + \underbrace{\mathbb{E}_{x'\sim\rho_h}\left[\|\xi(\kappa(x'))-\hat{\xi}(\kappa(x'))\|_1\right]}_{\text{Term 3}}$$

We bound these terms separately below:

Term 1 Is bounded by $2\sqrt{m^3 e}$ using Lemma 15.

Term 2 For any $x' \in \mathcal{X}_h$, we have $\phi_h^\star(x') = \phi_h^\star(\kappa(x'))$ whenever κ is defined. As ξ depends upon $f^\star$ therefore, $\xi(x') = \xi(\kappa(x'))$. This means Term 2 is 0.

Term 3 Fix $u > 0$ then using Lemma 16 we have:

$$\mathbb{E}_{x'\sim\rho_h}\left[\|\xi(\kappa(x'))-\hat{\xi}(\kappa(x'))\|_1\right] \leq \mathbb{E}_{x'\sim\rho_h}\left[\max_{i\in[m]}\|\xi(x'^{(i)})-\hat{\xi}(x'^{(i)})\|_1\right] \leq u.$$

Plugging this gives a bound of $2\sqrt{m^3 e} + u + b_j\tau$. The failure probability due to Lemma 18 is δ_c, due to Lemma 14 is $\frac{m\Delta_{cerr}(n,\delta)}{e}$ and due to Lemma 16 is $\frac{2\sqrt{m^5 e}}{u}$. This gives us success probability of at least $1-\delta_c-\frac{m\Delta_{cerr}(n,\delta)}{e}-\frac{2\sqrt{m^5 e}}{u}$. □

We want to make sure that there are only a small number of clusters. Let $\mathrm{N}_{\mathrm{BD}}^{(h)}$ be the number of backward kinematically inseparable states at timestep h. We know $\mathrm{N}_{\mathrm{BD}}^{(h)} \leq |\mathcal{S}|$. If x'_1 and x'_2 come from backward kinematically inseparable states then for any $x \in \mathcal{X}, a \in \mathcal{A}$ we have $f^\star(x,a,x'_1) = f^\star(x,a,x'_2)$. This further implies $\xi(x'_1) = \xi(x'_2)$. Therefore, we would hope that the number of clusters are at most $\mathrm{N}_{\mathrm{BD}}^{(h)}$ with high probability. We show this with the next result.

Lemma 21 (Number of Clusters are Small)**.** *If $\tau \geq 2u$ then under success of Lemma 16, we have $k \leq N_{BD}^{(h)}$ i.e.,* `GreedyClustering` *subroutine outputs at most $N_{BD}^{(h)}$ clusters.*

Proof. Say $x'_1, x'_2 \in \mathcal{U}$ come from backward kinematically inseparable states then $\xi(x'_1) = \xi(x'_2)$. Therefore, from triangle inequality and Lemma 16 we have:

$$\|\hat{\xi}(x'_1)-\hat{\xi}(x'_2)\|_1 \leq \|\hat{\xi}(x'_1)-\xi(x'_1)\|_1 + \|\xi(x'_2)-\hat{\xi}(x'_2)\|_1 \leq 2u.$$

Observe that this holds for any pair of observations in $\mathcal{U}$ that are backward kinematically inseparable, with success probability of Lemma 16.

Consider the behaviour of `GreedyClustering` when $\tau \geq 2u$. The moment we pick any vector $\hat{\xi}(x'_1)$, we are assured to have all remaining observations in $\mathcal{Z}$ that are backward kinematically inseparable to x'_1 in $\mathcal{B}_\tau(\hat{\xi}(x'_1))$ (Algorithm 4, line 5). These observations will then be removed from further consideration. Therefore, the algorithm can output at most as many clusters as there are backward kinematically inseparable states which is $\mathrm{N}_{\mathrm{BD}}^{(h)}$. It can however, output much fewer clusters if we pick τ to be too large (e.g., picking $\tau \geq m$ leads to 1 cluster). The proof is completed by observing that the only probabilistic statement we assumed is Lemma 16. □

E.2. Main Result

Theorem 9 (Main Theorem)**.** *Assume $m \geq \frac{2N|\mathcal{A}|}{\alpha\eta_{min}} \ln\left(\frac{eN^2|\mathcal{A}|}{\delta}\right)$ and $\tau = \frac{8m^3\sqrt{\Delta_{cerr}(n,\delta)}}{\delta^{3/2}}$. Then we have:*

$$\mathbb{E}_{x,a,x'_1,x'_2\sim D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\,|f^\star(x,a,x'_1) - f^\star(x,a,x'_2)|\right] \leq 32m^3\sqrt{\frac{\Delta_{cerr}(n,\delta)}{\delta^3}}, \tag{11}$$

and there are at most N abstract states. Both claims hold with probability at least $1-5\delta$.

Proof. For any x'_1, x'_2 we have from triangle inequality:

$$\|\xi(x'_1) - \xi(x'_2)\|_1 \leq \|\xi(x'_1) - \hat{\xi}(x'_1)\|_1 + \|\hat{\xi}(x'_1) - \hat{\xi}(x'_2)\|_1 + \|\hat{\xi}(x'_2) - \xi(x'_2)\|_1 \tag{12}$$

Multiplying by $\mathbf{1}\{\hat{\phi}(x'_1) = \hat{\phi}(x'_2)\}$ on both sides and taking expectation with respect to ρ_h we get:

$$\begin{aligned}
\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}(x'_1) = \hat{\phi}(x'_2)\}\|\xi(x'_1) - \xi(x'_2)\|_1\right] \quad \leq \quad & \underbrace{\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\xi(x'_1) - \hat{\xi}(x'_1)\|_1\right]}_{\text{Term 1}} \\
& + \underbrace{\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\hat{\xi}(x'_1) - \hat{\xi}(x'_2)\|_1\right]}_{\text{Term 2}} \\
& + \underbrace{\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\hat{\xi}(x'_2) - \xi(x'_2)\|_1\right]}_{\text{Term 3}}
\end{aligned}$$

It is easy to see that Term 1 and Term 3 are the same. We bound these below.

Bounding Term 1 and Term 3 : We bound these terms using $\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\} \leq 1$ and Lemma 15.

$$\mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\xi(x'_1) - \hat{\xi}(x'_1)\|_1\right] \leq \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\|\xi(x'_1) - \hat{\xi}(x'_1)\|_1\right] \leq 2\sqrt{m^3 e}$$

Bounding Term 2 :

$$\begin{aligned}
& \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\|\hat{\xi}(x'_1) - \hat{\xi}(x'_2)\|_1\right] \\
= \quad & \sum_j \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = j\}\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_2) = j\}\|\hat{\xi}(x'_1) - c_j + c_j - \hat{\xi}(x'_2)\|_1\right] \\
\leq \quad & \sum_j \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = j\}\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_2) = j\}\|\hat{\xi}(x'_1) - c_j\|_1\right] + \\
& \sum_j \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = j\}\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_2) = j\}\|c_j - \hat{\xi}(x'_2)\|_1\right], \qquad \text{(using triangle inequality)} \\
= \quad & 2\sum_j \mathbb{E}_{x'_1,x'_2\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = j\}\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_2) = j\}\|\hat{\xi}(x'_1) - c_j\|_1\right] \\
= \quad & 2\sum_j b_j \mathbb{E}_{x'_1\sim\rho_h}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = j\}\|\hat{\xi}(x'_1) - c_j\|_1\right], \qquad \text{(using definition of } b_j) \\
\leq \quad & 2\sum_j b_j \left\{2\sqrt{m^3 e} + u + b_j\tau\right\}, \qquad \text{(using Lemma 20)} \\
= \quad & 4\sqrt{m^3 e} + 2u + \tau\sum_j b_j^2 \\
\leq \quad & 4\sqrt{m^3 e} + 2u + \tau, \qquad \text{(using } \textstyle\sum_j b_j = 1 \text{ and } b_j \geq 0)
\end{aligned}$$

We plug the bounds for these terms in the triangle inequality result. Further, the left hand side of the triangle inequality can be lower bounded using Lemma 19. This gives us:

$$\mathbb{E}_{x,a,x'_1,x'_2\sim D_{\text{coup}}}\left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x'_1) = \hat{\phi}_h^{(\mathrm{B})}(x'_2)\}\,|f^\star(x,a,x'_1) - f^\star(x,a,x'_2)|\right] \leq \frac{1}{d}\left\{8\sqrt{m^3 e} + 2u + \tau\right\}.$$

The proof is completed by noting that the total failure probability from union bound is $\delta + 2\delta_c + \frac{m\Delta_{cerr}(n,\delta)}{e} + \frac{2\sqrt{m^5 e}}{u}$ which consists of: δ due to Theorem 8, $\frac{m\Delta_{cerr}(n,\delta)}{e}$ due to Lemma 15, failure probability due to Lemma 16 in addition to that of Lemma 15 is $\frac{2\sqrt{m^5 e}}{u}$ and combined failure probability due to Lemma 18 is and Lemma 17 is $2\delta_c$.

Remaining step is to set the hyperparameters. We set them as follows (not optimal necessarily) which gives the desired error bound along with lower bound on m using Equation 8.

$$d = 1; \qquad \delta_c = \delta; \qquad e = \frac{m\Delta_{cerr}(n,\delta)}{\delta} \Rightarrow \frac{m\Delta_{cerr}(n,\delta)}{e} = \delta;$$
$$u = \frac{4m^3\sqrt{\Delta_{cerr}(n,\delta)}}{\delta^{3/2}} \Rightarrow \frac{2\sqrt{m^5 e}}{u} \leq \delta, \qquad \tau = 2u \Rightarrow \tau = \frac{8m^3\sqrt{\Delta_{cerr}(n,\delta)}}{\delta^{3/2}}$$

Lastly, as we assume $\tau = 2u$ therefore, from Lemma 21 we have at most $\mathrm{N}_{\mathrm{BD}}^{(h)}$ many clusters (or equivalently learned abstract states) which we assume is less than N (the input to the algorithm). □

Wrapping up the proof The main theorem almost gets us to Lemma 10. If we can bridge this gap then we can reuse the rest of the analysis from Appendix D. We show how to get the same left hand side as Lemma 10 below.

Recall the definition of $V : \mathcal{X}_h \times \mathcal{X}_h \times \mathcal{X}_{h-1} \times \mathcal{A} \to \mathbb{R}$ from Appendix D given below:

$$V(x_1', x_2', x, a) := \frac{T(g^\star(x_1') \mid g^\star(x), a)}{\rho_h(g^\star(x_1'))} - \frac{T(g^\star(x_2') \mid g^\star(x), a)}{\rho_h(g^\star(x_2'))}.$$

Using the structure of $f^\star$ from Lemma 9 we can easily show:

$$|f^\star(x, a, x_1') - f^\star(x, a, x_2')| = \frac{\rho_h(g^\star(x_1'))}{T(g^\star(x_1') \mid g^\star(x), a) + \rho_h(g^\star(x_1'))} \cdot \frac{\rho_h(g^\star(x_2'))}{T(g^\star(x_2') \mid g^\star(x), a) + \rho_h(g^\star(x_2'))} |V(x_1', x_2', x, a)|$$

Using the lower bound on ρ_h from Lemma 8 and using $T(g^\star(x') \mid g^\star(x), a) + \rho_h(g^\star(x')) \leq 2$ then gives us:

$$|f^\star(x, a, x_1') - f^\star(x, a, x_2')| \geq \left(\frac{\alpha\eta_{min}}{2N|\mathcal{A}|}\right)^2 |V(x_1', x_2', x, a)|$$

Plugging this inequality in Theorem 9 gives us the following with probability of at least $1 - 5\delta$:

$$\mathbb{E}_{x,a,x_1',x_2' \sim D_{\text{coup}}} \left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = \hat{\phi}_h^{(\mathrm{B})}(x_2')\} |V(x_1', x_2', x, a)|\right] \leq 32m^3\sqrt{\frac{\Delta_{cerr}(n,\delta)}{\delta^3}} \left(\frac{2N|\mathcal{A}|}{\alpha\eta_{min}}\right)^2.$$

For any $i \in [N]$, we trivially have $\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i = \hat{\phi}_h^{(\mathrm{B})}(x_2')\} \leq \mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = \hat{\phi}_h^{(\mathrm{B})}(x_2')\}$, which gives us:

$$\mathbb{E}_{x,a,x_1',x_2' \sim D_{\text{coup}}} \left[\mathbf{1}\{\hat{\phi}_h^{(\mathrm{B})}(x_1') = i = \hat{\phi}_h^{(\mathrm{B})}(x_2')\} |V(x_1', x_2', x, a)|\right] \leq 32m^3\sqrt{\frac{\Delta_{cerr}(n,\delta)}{\delta^3}} \left(\frac{2N|\mathcal{A}|}{\alpha\eta_{min}}\right)^2.$$

Now we have the same left hand side as in Lemma 10, therefore, we can proceed the same way as we did in Appendix D. The only difference is that we have a different right hand side now. However, this only effects the final guarantees and not the proof technique which is identical to Appendix D from here. The final guarantees we obtain are still polynomial but worse than in Theorem 1. We omit repeating the technical steps for brevity.

F. Supporting results

The next lemma is the well-known performance difference lemma, which has appeared in much prior work (Bagnell et al., 2004; Kakade, 2003; Ross & Bagnell, 2014; Dann et al., 2017). Our version, which is adapted to episodic problems, is most closely related to Lemma 4.3 of (Ross & Bagnell, 2014). We provide a short proof for completeness.

Lemma 22 (Performance difference lemma)**.** *For any episodic decision process with any reward function R, and any two non-stationary policies $\pi_{1:H}^{(1)}$ and $\pi_{1:H}^{(2)}$, let $Q_h^{(1)} \in \Delta(\mathcal{X}_h)$ be the distribution at time h induced by policy $\pi_{1:H}^{(1)}$. Then we have*

$$V(\pi_{1:H}^{(1)}; R) - V(\pi_{1:H}^{(2)}) = \sum_{h=1}^{H} \mathbb{E}_{x_h \sim Q_h^{(1)}} \left[V(x_h; \pi_h^{(1)} \circ \pi_{h+1:H}^{(2)}) - V(x_h; \pi_{h:H}^{(2)})\right].$$

Proof. The proof is a standard telescoping argument.

$$\begin{aligned} V(\pi^{(1)}_{1:H}; R) - V(\pi^{(2)}_{1:H}) &= V(\pi^{(1)}_{1:H}; R) - V(\pi^{(1)}_1 \circ \pi^{(2)}_{2:H}; R) + V(\pi^{(1)}_1 \circ \pi^{(2)}_{2:H}; R) - V(\pi^{(2)}_{1:H}) \\ &= V(\pi^{(1)}_1 \circ \pi^{(2)}_{2:H}; R) - V(\pi^{(2)}_{1:H}) + \mathbb{E}_{x_2 \sim Q_2^{(1)}} \left[V(\pi^{(1)}_{2:H}; R) - V(\pi^{(2)}_{2:H}; R) \right]. \end{aligned}$$

The result follows by repeating this argument on the second term. □

The next result is Bernstein's inequality. The proof can be found in a number of textbooks (c.f., Boucheron et al., 2013).

Proposition 10 (Bernstein's inequality)**.** *If $U_1, \ldots, U_n$ are independent zero-mean random variables with $|U_t| \le R$ a.s., and $\frac{1}{n}\sum_{t=1}^n \mathrm{Var}(U_t) \le \sigma^2$, then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ we have*

$$\frac{1}{n}\sum_{t=1}^n U_t \le \sqrt{\frac{2\sigma^2 \ln(1/\delta)}{n}} + \frac{2R\ln(1/\delta)}{3n}.$$

The next proposition concerns learning with square loss, using a function class $\mathcal{G}$ with parametric metric entropy growth rate. Let D be a distribution over (x, y) pairs, where $x \in \mathcal{X}$ is an example space and $y \in [0,1]$. With a sample $\{(x_i, y_i)\}_{i=1}^n \overset{iid}{\sim} D$ and a function class $\mathcal{G} : \mathcal{X} \to [0,1]$, we may perform empirical risk minimization to find

$$\hat{g} := \operatorname*{argmin}_{g \in \mathcal{G}} \hat{R}_n(g) := \operatorname*{argmin}_{g \in \mathcal{G}} \frac{1}{n}\sum_{i=1}^n (g(x_i) - y_i)^2. \tag{13}$$

The population risk and minimizer are defined as

$$g^\star := \operatorname*{argmin}_{g \in \mathcal{G}} R(g) := \operatorname*{argmin}_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim D}(g(x) - y)^2.$$

We assume *realizability*, which means that the Bayes optimal classifier $x \mapsto \mathbb{E}_D[y \mid x]$ is in our class, and as this minimizes the risk over all functions we know that $g^\star(x)$ is precisely this classifier.

We assume that $\mathcal{G}$ has "parametric" pointwise metric entropy growth rate, which means that the pointwise covering number at scale ε, which we denote $\mathcal{N}(\mathcal{G}, \varepsilon)$ scales as $\mathcal{N}(\mathcal{G}, \varepsilon) \le c_0 d \ln(1/\varepsilon)$, for a universal constant $c_0 > 0$. Recall that for a function class $\mathcal{G} : \mathcal{X} \to \mathbb{R}$ the pointwise covering number at scale ε is the size of the smallest set $V : \mathcal{X} \to \mathbb{R}$ such that

$$\forall g \in \mathcal{G}, \exists v \in V : \sup_x |g(x) - v(x)| \le \varepsilon.$$

With the above definitions, we can state the main guarantee for the empirical risk minimizer.

Proposition 11. *Fix $\delta \in (0,1)$. Let $\hat{g}$ be the empirical risk minimizer in (13) based on n samples from a distribution D. If $\mathcal{G}$ is realizable for D and has parametric entropy growth rate, then with probability at least $1-\delta$ we have*

$$\mathbb{E}_{(x,y)\sim D}\left[(\hat{g}(x) - g^\star(x))^2 \right] \le \Delta_{reg}, \text{ with } \Delta_{reg} := \inf_{\varepsilon > 0} \left\{ 6\varepsilon + \frac{8\ln(2\mathcal{N}(\mathcal{G}, \varepsilon)/\delta)}{n} \right\}.$$

The result here is a standard square loss excess risk bound, and it is perhaps the simplest such bound for well-specified infinite function classes. Sharper guarantees based on using empirical covering numbers, combinatorial parameters (Alon et al., 1997), or localization (Liang et al., 2015), are possible, and completely composable with the rest of our arguments. In other words, if a bound similar to the one in Proposition 11 is achievable under different assumptions (e.g., different measure of statistical complexity), we can incorporate it into the proof of Theorem 1. We state, prove, and use this simple bound to keep the arguments self contained.

Proof. Define

$$Z_i(g) = (g(x_i) - y_i)^2 - (g^\star(x_i) - y_i)^2.$$

Using the realizability assumption that $\mathbb{E}[y \mid x] = g^\star(x)$, it is easy to verify that

$$\mathbb{E}[Z_i(g)] = \mathbb{E}[g(x)^2 - g^\star(x)^2 - 2y(g(x) - g^\star(x))] = \mathbb{E}[(g(x) - g^\star(x))^2].$$

The variance is similarly controlled:

$$\begin{aligned}\mathrm{Var}[Z_i(g)] &\le \mathbb{E}[Z_i(g)^2] = \mathbb{E}[(g(x) + g^\star(x) - 2y)^2(g(x) - g^\star(x)^2] \\ &\le 4\mathbb{E}[(g(x) - g^\star(x)^2] = 4\mathbb{E}[Z_i(g)],\end{aligned}$$

where we use that $y, g(x), g^\star(x)$ are in $[0, 1]$. Therefore, via Bernstein's inequality (Proposition 10), with probability at least $1 - \delta$ we have

$$\left|\frac{1}{n}\sum_i Z_i(g) - \mathbb{E}[Z(g)]\right| \le 2\sqrt{\frac{\mathbb{E}[Z(g)]\ln(2/\delta)}{n}} + \frac{2\ln(2/\delta)}{n}. \tag{14}$$

For the uniform convergence step, we show that $Z_i(g)$ is a Lipschitz function in g:

$$|Z_i(g) - Z_i(g')| = \left|(g(x_i) - g'(x_i))^2(g(x_i) + g'(x_i) - 2y_i)^2\right| \le 4\left|g(x_i) - g'(x_i)\right|,$$

where we again use that $y_i, g(x_i)$ and $g'(x_i)$ are in $[0, 1]$.

Now let V be a pointwise cover of $\mathcal{G}$ at scale ε, so that for any $g \in \mathcal{G}$ there exits $v_g \in V$ such that:

$$\sup_x |v_g(x) - g(x)| \le \varepsilon.$$

By our metric entropy assumptions, we have that $|V| \le \mathcal{N}(\mathcal{G}, \varepsilon) \le \varepsilon^{-c_0 d}$. For any $g \in \mathcal{G}$ we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n Z_i(g) &\le \varepsilon + \frac{1}{n}\sum_{i=1}^n Z_i(v_g) \le \varepsilon + \mathbb{E}[Z(v_g)] + 2\sqrt{\frac{\mathbb{E}[Z(v_g)]\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n}} + \frac{2\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n} \\ &\le \varepsilon + 2\mathbb{E}[Z(v_g)] + \frac{3\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n} \le 3\varepsilon + 2\mathbb{E}[Z(v_g)] + \frac{3\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n}.\end{aligned}$$

Here we are applying (14) uniformly over the pointwise cover, using the fact that $2\sqrt{ab} \le a + b$, and using the pointwise covering property. Similarly we can control the other tail

$$\begin{aligned}\mathbb{E}[Z(g)] &\le \varepsilon + \mathbb{E}[Z(v_g)] \le \varepsilon + \frac{1}{n}\sum_{i=1}^n Z_i(v_g) + 2\sqrt{\frac{\mathbb{E}[Z(v_g)]\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n}} + \frac{2\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n} \\ &\le \frac{5}{2}\varepsilon + \frac{1}{n}\sum_{i=1}^n Z_i(g) + \frac{1}{2}\mathbb{E}[Z(g)] + \frac{4\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n}\end{aligned}$$

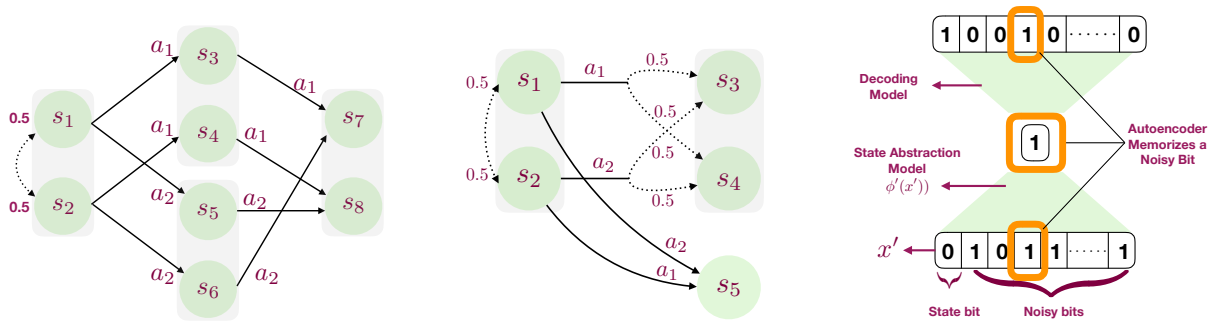
Re-arranging and putting the two bounds together, the following holds simultaneously for all $g \in \mathcal{G}$, with probability at least $1 - \delta$

$$\frac{1}{2}\mathbb{E}[Z(g)] - 3\varepsilon - \frac{4\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n} \le \frac{1}{n}\sum_{i=1}^n Z_i(g) \le 2\mathbb{E}[Z(g)] + 3\varepsilon + \frac{4\ln(2\mathcal{N}(\mathcal{G},\varepsilon)/\delta)}{n}. \qquad \square$$

G. Can We Use Existing State Abstraction Oracles?

Our analysis verifies the utility of the backward KI state abstraction: it enables efficient reward-free exploration and it can be learned using contrastive estimation procedure as shown with HOMER. Do other, previously studied, state abstractions admit similar properties?

In this section, we discuss prior approaches for learning state abstractions. In Block-MDPs, we show that these approaches fail to find a policy cover when interleaved with a PSDP-style routine used to find policies that visit the abstract states, following the structure of HOMER. Note that it may be possible to embed these approaches in other algorithmic frameworks and successfully explore.



(a) (b) (c)

Figure 5: Counterexamples for prior work on abstraction/representation learning. We do not show observations for brevity. **Left:** A Block MDP where predicting the previous action from observations (Pathak et al., 2017) or predicting the previous abstract state and action fails (Du et al., 2019). **Middle:** A Block-MDP where the model-based algorithm of Du et al. (2019) fails. **Right:** Illustration of a failure mode for the autoencoding approach of Tang et al. (2017), where optimal reconstruction loss is attained by memorizing noise. See text for more details.

Predicting Previous Action from Observations. Curiosity-based approaches learn a representation by predicting the previous action from the previous and current observation (Pathak et al., 2017). When embedded in a PSDP-style routine, this approach fails to guarantee coverage of the state space, as can be seen in Figure 5a. A Bayes optimal predictor of previous action a given previous and current observations x, x' collapses the observations generated from $\{s_3, s_4\}$, $\{s_5, s_6\}$, and $\{s_7, s_8\}$ together. To see why, the agent can only transition to $\{s_3, s_4\}$ by taking action a_1, so we can perfectly predict the previous action even if all of the observations from these states have the same representation. This also happens with $\{s_5, s_6\}$ and $\{s_7, s_8\}$. Unfortunately, collapsing observations from $\{s_7, s_8\}$ together creates an unfavorable tie-breaking scenario when trying to find a policy to visit this representation. For example, the policy that takes action a_1 in s_1 and s_3 and a_2 in s_2 and s_6 deterministically reaches s_7, so it visits this representation maximally, but it never visits s_8. So this approach for representation learning, interleaved with policy optimization, does not yield a policy cover.

Predicting Previous Action and Abstract State. Instead of predicting the previous action, Du et al. (2019) learn a representation by predicting the previous action *and* previous abstract state. As they show, this approach provably explores a restricted class of Block-MDPs, but unfortunately it fails in the general case. For example in Figure 5a, a Bayes optimal predictor collapses observations from $\{s_1, s_2\}, \{s_3, s_4\}, \{s_5, s_6\}$, and $\{s_7, s_8\}$, leading to the same failure for policy optimization as the curiosity-based approach. This state collapse is caused by a stochastic start state; $\{s_1, s_2\}$ cannot be separated by this approach and using the joint representation for $\{s_1, s_2\}$ as a prediction target causes a cascading failure. Note that Du et al. (2019) assume a deterministic start state in their analysis.

Instead of a PSDP-style routine, Du et al. (2019) use a model-based approach for building a policy cover, where the learned policies operate directly on the abstract states. Actually this approach avoids the tie-breaking issue in policy optimization and does succeed for the example in Figure 5a, but it fails in Figure 5b. If policies are defined over abstract states, we must take the same action in s_1 and s_2 (as this approach can never separate a stochastic start state), so we can reach $\{s_3, s_4\}$ with probability at most $1/2$, while a policy operating directly on observations could reach these states with probability 1. Chaining this construction together shows that this approach can at best find an α-policy cover where α is exponentially small in the horizon.

Training Autoencoders. The final approach uses an autoencoder to learn a representation, similar to Tang et al. (2017). Here we representation ϕ and decoder U by minimizing reconstruction loss $\texttt{dist}(x, U(\phi(x)))$ over a training set of raw observations, where $\texttt{dist}$ is a domain-specific distance function. Figure 5c shows that this approach may fail to learn a meaningful representation altogether. The problem contains just two states and the observations are d-dimensional binary vectors, where the first bit encodes the state and the remaining bits are sampled from $\text{Ber}(1/2)$ (it is easy to see that this is a Block-MDP). For this problem, optimal reconstruction loss is achieved by a representation that ignores the state bit and memorizes the noise. For example, if ϕ has a single output bit (which suffices as there are only two states), it is never worse to preserve a noise bit than the state bit. In fact, if one state is more probable than the other, then predicting a noise bit along with the most likely state results in *strictly* better performance than predicting the state bit. So a representation using this

approach can ignore state information and is not useful for exploration.

Bisimulation. A number of other abstraction definitions have been proposed and studied in the state abstraction literature (c.f., (Givan et al., 2003; Li et al., 2006)). The finest definition typically considered is *bisimulation* or *model-irrelevance abstraction*, which aggregates two observations x_1, x_2 if they share the same reward function and the same transition dynamics over the abstract states, e.g., for each abstract state s', $T(\phi(x') = s' \mid x_1, a) = T(\phi(x') = s' \mid x_2, a)$, where ϕ is the abstraction. A natural reward-free notion simply aggregates states if they share the same dynamics over abstract states, ignoring the reward condition. There are two issues with using bisimulations and, as a consequence, coarser abstraction notions. First, the trivial abstraction that aggregates all observations together is a reward-free bisimulation, which is clearly unhelpful for exploration. More substantively, learning a reward-sensitive bisimulation is statistically intractable, requiring a number of samples that is exponential in horizon (Proposition B.1 in Modi et al. (2019)).

An even finer definition than bisimulation, which has appeared informally in the literature, aggregates two observations if they share the same reward function and the same transition dynamics over the observations (Equation 2 in Jiang (2018)). The reward-free version is equivalent to forward kinematic inseparability. However, we are not aware of any prior work that attempts to learn such an abstraction, as we do here.

Summary. These arguments show that previously studied state-abstraction or representation learning approaches cannot be used for provably efficient exploration in general Block-MDPs, at least when used with a HOMER-like algorithm. We emphasize that our analysis does not preclude the value of these approaches in other settings (e.g., outside of Block-MDPs) or when used in other algorithms. Moreover, the remarks here are of a worst case nature and do not necessarily imply that the approaches are empirically ineffective.

H. Experimental Setup, Optimization Details and Additional Results

Emission Process in Diabolical Combination Lock 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 as follows. First, the current state information (whether it is of type $s_{h,a}$, $s_{h,b}$ or $s_{h,c}$) and time step (h) are encoded into one-hot vectors which are concatenated together and added with an isotropic Gaussian vector with mean 0 and variance 0.1. This vector is then padded with an all-zero vector to lift into d dimension and finally multiplied by Hadamard matrix of order d. A Hadamard matrix of order d, denoted H_d, is a $d \times d$ matrix with entries in $\{-1, +1\}$ and mutually orthogonal rows. As d is a power of 2, hence we can construct H_d using Sylvester's method which defines $H_l = \begin{bmatrix} H_{\frac{l}{2}} & H_{\frac{l}{2}} \\ H_{\frac{l}{2}} & -H_{\frac{l}{2}} \end{bmatrix}$ for any $l \in \mathbb{N}$ and $H_1 = [1]$.

Note that diabolical combination lock is not strictly a Block MDP setting as the same observation can be emitted from two states although with a small probability. Our experiments, therefore, test the resilience of our results to small violation.

Problem Figure We visualize the diabolical combination lock problem in Figure 6.

Modeling Details for HOMER. As stated before, 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 want to recover a forward and a backward abstraction from the model class $\mathcal{F}_N$ using the REG oracle. However, our theoretical results never assume that these are recovered from the same model. For example, the analysis of learned backward abstraction $\hat{\phi}_h^{(\text{B})}$ in Appendix D does not use the fact that the model $\hat{f}$ also has a forward abstraction $\hat{\phi}_{h-1}^{(\text{F})}$. This allows us to train two different models with single bottleneck, for recovering forward and backward abstraction separately. Empirically, training a model with a single bottleneck is easier than a model with two bottlenecks. We implement these two model class in a similar way barring the place where we place the bottleneck.

We now describe the model details for recovering $\hat{\phi}_h^{(\text{B})} : \mathcal{X} \to [N]$. We represent the state abstraction function $\hat{\phi}_h^{(\text{B})} : \mathcal{X} \to [N]$ using a linear model $B \in \mathbb{R}^{N \times d}$ with $\hat{\phi}_h^{(\text{B})}(x') = \arg\max_{i \in [N]} (Bx')_i$. Given a tuple (x, a, x') we form a vector by concatenating Ax, $\mathbf{1}_a$, and z together, where $A \in \mathbb{R}^{M \times d}$, $\mathbf{1}_a$ is the one-hot encoding of the action, and $z_i \propto \exp((Bx')_i + g_i)$ applies the Gumbel softmax trick (Jang et al., 2016) to convert Bx' into a probability distribution (g_i is an independent Gumbel random variable). Then we pass the concatenated vector into a two layer feed-forward neural network with leaky rectified linear units (Maas et al., 2013) and a softmax output node to obtain the prediction. We generate softmax

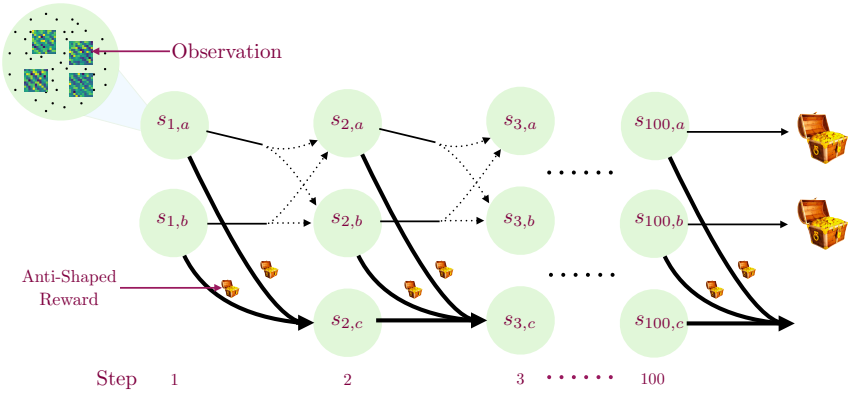


Figure 6: Illustrates the diabolical combination lock problem which contains multiple challenges including sparse anti-shaped rewards, rich-observations, long horizons and extremely sparse good rewards. We do not visualize the observations for every state for brevity.

probabilities instead of a scalar value as we train the model using cross-entropy loss instead of squared loss (described later). The parameters of the model include the weight matrices A, B that form the abstraction, as well as the parameters of the feed-forward neural network.

The model class that recovers $\hat{\phi}^{(\mathrm{F})}_{h-1}$ are similar to the one for recovering $\hat{\phi}^{(\mathrm{B})}_{h}$. The only difference is that for a given tuple (x, a, x') we apply the Gumbel softmax trick to Ax instead of Bx'.We allow the capacity of the forward and backward abstractions to be different, i.e., M and N can be different.

Efficient Implementation of HOMER We make a few empirical changes to make HOMER more practically efficient. This includes changes to improve the computational-complexity and sample-efficiency.

Computationally-Efficient Implementation of HOMER. As stated, the most computationally expensive component of HOMER is the $\mathcal{O}(NH)$ calls to PSDP for learning the policy covers. This has a total computational cost of $\mathcal{O}(NH^3 n_{\mathrm{psdp}} + \mathrm{Time}_{\mathrm{pol}}(n_{\mathrm{psdp}})NH^2)$, but in practice, it can be significantly reduced. Empirically, we use two important optimizations: First, we parallelize the N calls to PSDP for optimizing the internal reward functions in each iteration of the algorithm (Algorithm 1, line 12–line 15). Second and perhaps more significantly, we can attempt to find compositional policies using a greedy search procedure (GPS). Here, rather than perform full dynamic programming to optimize reward $R_{i,h}$, we find the policy $\hat{\pi}_h$ for the last time step, and then we append this policy to the best one from our cover Ψ_{h-1}. Formally, we compute $\hat{\pi}_{1:h-1} = \mathrm{argmax}_{\pi \in \Psi_{h-1}} V(\pi \circ \hat{\pi}_h; R_{i,h})$, where $V(\cdot; R)$ is the value function with respect to reward function R and $\circ$ denotes policy composition. Then, since the optimal value with respect to $R_{i,h}$ is at most 1, we check if $V(\hat{\pi}_{1:h-1} \circ \hat{\pi}_h; R_{i,h}) \geq 1 - \epsilon$. If it is we need not perform a full dynamic programming backup, otherwise we revert to PSDP. GPS may succeed even though the policies we are trying to find are non-compositional in general. In the favorable situation where GPS succeeds, actually no further samples are required, since we can re-use the real transitions from the regression step, and we need only solve one contextual bandit problem, instead of H. Empirically, both of these optimizations may yield significant statistical and computational savings.

Statistically-Efficient Implementation of HOMER. The pseudocode stated in Algorithm 1 spends two episodes to create a single datapoint for the REG subroutine (Algorithm 1, line 5-line 10). This only effects our sample complexity bounds by a factor of 2 but in practice this is undesirable. Therefore, we use a more sample-efficient data collection procedure in our experiments. Firstly, we collect a set of n_{reg} i.i.d. observed transitions $\{(x_i, a_i, x'_i)\}_{i=1}^{n_{\mathrm{reg}}}$ using our sampling procedure (Algorithm 2, line 5), and we create imposter transitions by resampling within this set. The procedure to create imposter transitions is based on whether we are training a model class to recover $\hat{\phi}^{(\mathrm{B})}_{h}$ or $\hat{\phi}^{(\mathrm{F})}_{h-1}$. When training the model with bottleneck on x' (i.e., to recover $\hat{\phi}^{(\mathrm{B})}_{h}$), we create imposter transitions $\{(x_i, a_i, \tilde{x}'_i)\}_{i=1}^{n_{\mathrm{reg}}}$ where for each i, $\tilde{x}'_i$ is chosen uniformly from $\{x'_1, \ldots, x'_{n_{\mathrm{reg}}}\}$. The dataset for performing contrastive estimation is then given by $\{([x_i, a_i, x'_i], 1)\}_{i=1}^{n_{\mathrm{reg}}} \cup \{([x_i, a_i, \tilde{x}'_i], 0)\}_{i=1}^{n_{\mathrm{reg}}}$. Similarly, when we are training the model with bottleneck on x (i.e., to recover $\hat{\phi}^{(\mathrm{F})}_{h-1}$), we create imposter transitions $\{(\tilde{x}_i, \tilde{a}_i, x'_i)\}_{i=1}^{n_{\mathrm{reg}}}$ where where for each i, $\tilde{x}_i$ and $\tilde{a}_i$ are chosen uniformly from $\{x_1, \ldots, x_{n_{\mathrm{reg}}}\}$ and $\{a_1, \ldots, a_{n_{\mathrm{reg}}}\}$ respectively. The dataset for performing contrastive estimation, in this case, is given by $\{([x_i, a_i, x'_i], 1)\}_{i=1}^{n_{\mathrm{reg}}} \cup \{([\tilde{x}_i, \tilde{a}_i, x'_i], 0)\}_{i=1}^{n_{\mathrm{reg}}}$. We use the same

set of observed transitions to create the dataset for training both models.

Learning Details for HOMER. We describe the details of the oracle and hyperparameter below:

Implementing REG *oracle:* We implement the REG subroutine by performing supervised binary classification instead of regression. Formally, we train the model $f(x', a, x)$ on a training data $\{(x_i, a_i, x'_i, y_i)\}_{i=1}^{n_{\text{reg}}}$ as shown below:

$$\max_{f \in \mathcal{F}_N} \frac{1}{n_{\text{reg}}} \sum_{i=1}^{n_{\text{reg}}} \ln\{y_i f\left(x_i, a_i, x'_i\right) + (1 - y_i)(1 - f\left(x_i, a_i, x'_i\right)\}$$

We use Adam optimization with mini batches. We separate a validation set from the training data. We train the model for a maximum number of epochs and stop when the best validation performance doesn't improve for a certain fixed (patience) number of epochs. We use the model with the best validation performance. We found that learning is more stable if we initialize the model by first training without the quantization step. This is achieved by not performing the Gumbel softmax trick and directly using the underlying vector (e.g., using Bx' instead of z when learning $\hat{\phi}_h^{(\text{B})}$). The two training procedures are identical barring the quantization step in the model.

Implementing CB *oracle:* We learn policies for the CB subroutine by training a model to predict the immediate reward using mean squared loss error instead of performing cost-sensitive classification. This is equivalent to one-step Q-learning. Formally, we train a model $Q_\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ on training data $\{(x_i, a_i, p_i, r_i)\}_{i=1}^{n_{\text{psdp}}}$ as shown below:

$$\max_{\theta} \frac{1}{n_{\text{psdp}}} \sum_{i=1}^{n_{\text{psdp}}} \left(Q_\theta(x_i, a_i) - r_i\right)^2.$$

The policy corresponding to Q_θ deterministically takes the action $\arg\max_{a \in \mathcal{A}} Q_\theta(x, a)$. We use Adam optimization with mini batches. Similar to REG, we train for a maximum number of epochs and stop when the best validation performance does not improve for a certain fixed (patience) number of epochs. We use the model with best performance on the validation set.

The changes mentioned above do not change the key idea of HOMER which is to iterate between learning kinematic inseparability abstraction using a form of noise contrastive learning, and planning using the PSDP algorithm. These changes simply help to make the algorithm empirically more appealing by providing computational and statistical advantages.

We search for optimal hyperparameters using grid search. We search for n_{reg} over $\{5000, 10000\}$ and n_{psdp} over $\{10000, 15000, 20000\}$. We do not optimize other hyperparameters. For each hyperparameter setting, we run the algorithm five times with five different seeds. We use the median performance for finding the best hyperparameters. Hyperparameter values for the diabolical combination lock problem with $H = 100$ can be found in Table 3. We run the algorithm on GPU clusters containing Nvidia P100, V100 and 1080ti GPUs. It took approximately one day to run HOMER for $H = 100$.

We use the PyTorch library (version 1.1.0) for implementing HOMER.[5] We use the standard mechanism provided by PyTorch for initializing the parameters.

Learning Details for PCID. We use the code made publicly available by the authors.[6] PCID uses a model for predicting the previous state and action and performs k-means clustering on the predicted probabilities. We experimented with both linear models and feed-forward networks provided by authors. We optimized hyperparameters using grid search. We search for learning rate over $\{0.01, 0.05, 0.001\}$, and the data collection hyperparameter (n) used by their state decoding algorithm over $\{200, 1000, 10000\}$. The other hyperparameters were set to values recommended by the authors who evaluated on a combination lock problem similar to ours. We list the hyperparameter choice in Table 4.

Learning Details for PPO and PPO + RND. We train each baseline for a maximum of 10 million episodes. All baseline models use fully-connected, 2-layer MLPs with 64 hidden units and ReLU non-linearities. For each baseline, we used the RMSProp optimizer (Tieleman & Hinton, 2012) and tuned learning rates over $\{0.01, 0.001, 0.0001\}$. For PPO + RND, the random networks were 2-layer MLPs with 100 hidden units and ReLU non-linearities. We found that tuning the intrinsic reward coefficient λ_I was important to obtain good performance, and performed a search over $\lambda_I \in \{1, 10, 100, 1000, 10000\}$. We found that $\lambda_I = 1000$ performed best and used this value for all experiments. We experimented with applying a running normalization to the intrinsic reward as described in (Burda et al., 2019), but found that this did not improve over

[5] https://pytorch.org/

[6] https://github.com/Microsoft/StateDecoding

Table 3: HOMER Hyperparameters

Hyperparameter	Value
Learning Rate	0.001 (for both REG and CB)
Batch size	32 (for both REG and CB)
n_{reg}	We sample $10,000$ observed transitions from which we generate additional $10,000$ imposter transitions.
n_{psdp}	$20,000$
N (capacity of a backward state abstraction model)	2
M (capacity of a forward state abstraction model)	3
Maximum epochs for REG	200
Maximum epochs for CB	40
Validation data size (REG)	20% of the size of training data for REG.
Validation data size (CB)	20% of the size of training data for CB.
Hidden layer size for $\mathcal{F}_N$	56
Gumbel-Softmax temperature	1

Table 4: PCID Hyperparameters

Hyperparameter	Value
Learning Rate	0.01
n	200
Number of clusters for k-means	3

using the unnormalized intrinsic reward. We also experimented with higher entropy bonuses for PPO with $H = 6$, but this did not yield any improvement so we kept the default value of 0.01 for subsequent experiments. We used the PPO implementations provided in (Shangtong, 2018) and kept other hyperparameters fixed at their default values. We list the hyperparameter values for PPO and PPO + RND in Table 5. We found the best choice of hyperparameter was not dependent on H. All experiments were run on a cluster of Nvidia P100, V100 and 1080ti GPUs. Running the PPO + RND experiments for $H = 100$ took approximately 10 days.

Table 5: PPO and PPO + RND Hyperparameters

Hyperparameter	Value
Learning Rate	0.001
Rollout length	H
γ	0.99
τ_{GAE}	0.95
Gradient clipping	5
Entropy Bonus	0.01
Optimization Epochs	10
Minibatch size	160
Ratio clip	0.2
Extrinsic Reward coefficient λ_E	1.0
Intrinsic Reward coefficient λ_I (for PPO + RND)	1000

Learning Details for DQN. We used the OpenAI Baselines (Dhariwal et al., 2017) implementation of DQN. We tuned learning rates over the $\{0.01, 0.001, 0.0001\}$ and the ϵ-greedy exploration fraction over $\{0.01, 0.001, 0.0001\}$. The networks were 2-layer fully-connected MLPs with 64 hidden units. We also experimented with adding parameter noise, but this did not improve performance. All other hyperparameters were kept at their default values and are shown in Table 6.

Table 6: DQN Hyperparameters

Hyperparameter	Value
Learning Rate	0.001
Exploration Fraction	0.001
Replay Buffer Size	50000
Target Network Update Frequency	500
Prioritized Replay	true
Prioritized α	0.6
Prioritized β	0.4
Dueling	true
γ	0.99
Gradient clipping	10
Minibatch size	32

Algorithms	**Statistics**	$H = 3$	$H = 6$	$H = 12$	$H = 25$	$H = 50$	$H = 100$
PPO	Max	∞	∞	∞	∞	∞	∞
	Median	∞	∞	∞	∞	∞	∞
	Min	∞	∞	∞	∞	∞	∞
DQN	Max	1.69×10^4	∞	∞	∞	∞	∞
	Median	1.62×10^4	∞	∞	∞	∞	∞
	Min	1.58×10^4	∞	∞	∞	∞	∞
PPO + RND	Max	3.4×10^4	9.3×10^4	3.07×10^6	2.27×10^6	∞	∞
	Median	2.2×10^4	3.9×10^4	0.89×10^6	1.63×10^6	∞	∞
	Min	2×10^4	3.3×10^4	0.2×10^6	0.63×10^6	3.3×10^6	∞
PCID2019	Max	∞	∞	∞	∞	∞	∞
	Median	∞	∞	∞	∞	∞	∞
	Min	∞	∞	∞	∞	∞	∞
HOMER	Max	9.3×10^4	0.19×10^6	0.38×10^6	0.81×10^6	1.64×10^6	6.55×10^6
	Median	9.2×10^4	0.19×10^6	0.37×10^6	0.8×10^6	1.64×10^6	6.54×10^6
	Min	9.1×10^4	0.18×10^6	0.37×10^6	0.78×10^6	1.63×10^6	6.53×10^6

Table 7: Details of execution for each algorithm and horizon (H). The action space size is always 10. We run the algorithm 5 times with different seeds for every hyperparameter setting. We compute the number of episodes needed to achieve a mean return of $V(\pi^\star)/2$. If the algorithm fails to achieve this in 10^7 episodes then we report the result as ∞ denoting timeout. We report the median, min and max performance over different seeds corresponding to the best hyperparameter setting.

Additional Results

Results with error margin We show the results on the diabolical combination lock with error margins in Table 7.

Visitation Probabilities We visualize the visitation probabilities in Figure 7. We compare the performance of HOMER against the best baseline PPO + RND. For this experiment, we run PPO + RND on the setting $H = 100$ even though it failed for shorter horizons. The results show that HOMER maintains good coverage over every state unlike PPO + RND.

PPO + RND (fails to explore from time step 28)

HOMER (successfully explores at all time steps)

Figure 7: Visualization of the visitation probabilities for algorithms on the diabolical combination lock problem. The top row, middle row and the bottom row represent states in $\{s_{h,a}\}_{h=1}^{100}$, $\{s_{h,b}\}_{h=1}^{100}$ and $\{s_{h,c}\}_{h=1}^{100}$ respectively. The h^{th} column represents states reachable at time step h. We do not show observations or transition edges for brevity. We sample $100,000$ episodes uniformly through the execution of the algorithm and compute the number of time $\texttt{count}[s]$ the agent visits a state s. The count statistics is shown using the opacity of the fill of each state. Formally, we set opacity of s as $\propto \ln(\texttt{count}[s]+1)$. The more opaque the circles are the more frequently the agent visits them. HOMER is able to explore well for all time steps unlike the best baseline PPO + RND.