Learning Markov State Abstractions for Deep Reinforcement Learning

Cameron Allen*
Brown University
csal@cs.brown.edu

Neev Parikh
Brown University
neev_parikh@brown.edu

George Konidaris
Brown University
gdk@cs.brown.edu

Abstract

We introduce a method of learning an abstract state representation for Markov decision processes (MDPs) with rich observations. We begin by proving that a combination of three conditions is sufficient for a learned state abstraction to retain the Markov property. We then describe a practical training procedure that combines inverse model estimation and temporal contrastive learning to learn an abstraction that approximately satisfies these conditions. We evaluate our approach with a proof-of-concept visual gridworld experiment, where the learned representation captures the underlying structure of the domain and enables substantially improved learning performance over end-to-end deep RL, matching the performance achieved with hand-designed compact state information.

1 Introduction

Reinforcement learning (RL) in Markov decision processes with rich observations requires a suitable state representation. Typically, such representations are learned implicitly as a byproduct of doing deep RL. However, in domains where compact (non-visual) state information is available, an RL agent trained on compact states usually outperforms an agent trained on rich observations. Recent work has sought to close this "representation gap" by incorporating a wide range of representation-learning objectives, including pixel reconstruction (Finn et al., 2016; Higgins et al., 2017), pixel prediction (Song et al., 2016; Kaiser et al., 2020), abstract transition model estimation (Gelada et al., 2019), inverse model estimation (Christiano et al., 2016; Pathak et al., 2017), contrastive learning (Van den Oord et al., 2018; Anand et al., 2019; Misra et al., 2019; Srinivas et al., 2020; Stooke et al., 2020), and data augmentation (Laskin et al., 2020), as well as combinations of these objectives (Agrawal et al., 2016; Zhang et al., 2018; Shelhamer et al., 2016; Ha & Schmidhuber, 2018). Many of these methods improve performance, but most lack a theoretical justification for why they should be effective.

The fundamental assumption of reinforcement learning in Markov decision processes (MDPs) is that the MDPs are, in fact, Markov. This assumption calls for state representations that preserve the Markov property. We describe sufficient conditions for obtaining a Markov abstract state representation. We next show that these conditions are approximately satisfied using a combination of two popular representation learning objectives: inverse model estimation and temporal contrastive learning. Inverse model estimation predicts the action distribution that explains two consecutive states, and temporal contrastive learning determines whether two states were in fact consecutive. We evaluate this combined learning objective on a proof-of-concept representation learning problem, and show that it learns representations that capture the underlying structure of the domain and enable efficient learning. This work provides theoretical justification for why inverse models and temporal contrastive learning are each effective individually, and shows that their combination is uniquely suited to the goal of learning useful state representations for RL.

*Corresponding Author

Presented at Deep RL Workshop, NeurIPS 2020.

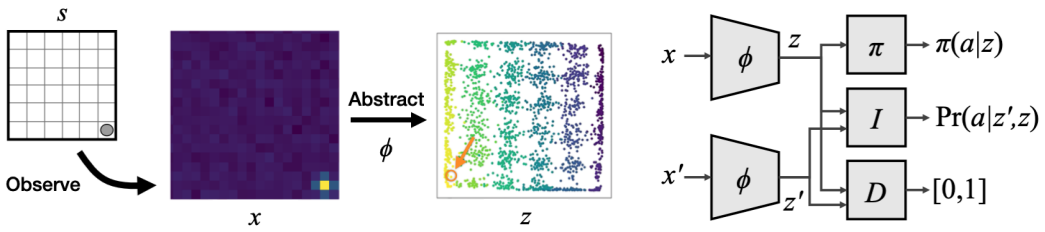


Figure 1: (a) A 6×6 visual gridworld domain where an abstraction function ϕ maps each high-dimensional observed state x to a lower-dimensional abstract state z (orange circle). (b) A shared encoder ϕ maps ground states x, x' to abstract states z, z', which are inputs to an inverse dynamics model I and a constrastive model D that discriminates between real and fake state transitions. The agent's policy depends only on the abstract state.

2 Background

2.1 Markov Decision Processes

A Markov decision process M consists of sets of states X and actions A, reward function $R : X \times A \times X \to \Re$, transition dynamics $T : X \times A \to \Pr(X)$, and discount factor γ. We use X as the name for the set of states, rather than S, which is more common, to highlight that X is a rich observation, perhaps generated by a noisy sensor function σ, but that the behavior of the MDP is in fact governed by much more compact (but unobserved) set of states S, as in the Block MDP formulation (Du et al., 2019). We assume that S and σ are unknown to the agent, and that the rich, observed state space X is itself Markov, which means it is a sufficient statistic for predicting the next state and expected reward, for any action the agent might select.

Definition 1 (Markov Property). *Given an decision process $M = (X, A, R, T, \gamma)$, the state space X is* Markov *if and only if the following holds, for all $a \in A$, $x \in X$, $k \geq 1$:*

$$\begin{aligned} \textit{Markov}(X) : \iff \quad & T^{(k)}\left(x_{t+1} | \{a_{t-i}, x_{t-i}\}_{i=0}^{k}\right) = T(x_{t+1} | a_t, x_t) \\ \cap \quad & R^{(k)}\left(x_{t+1}, \{a_{t-i}, x_{t-i}\}_{i=0}^{k}\right) = R(x_{t+1}, a_t, x_t). \end{aligned} \tag{1}$$

The superscript (k) denotes that the function is being conditioned on k additional steps of history. Technically, each $x \in X$ must also be sufficient for determining the set of actions available to the agent in that state, but here we assume, as is common, that this set is equal to A in every state.

The behavior of an RL agent is typically determined by a (Markov) policy $\pi : X \to \Pr(A)$, and each policy induces value function $V^\pi : X \to \Re$ (and action-value function $Q^\pi : X \times A \to \Re$), which estimates the sum of expected future discounted rewards starting from a given state (and action) and following the policy π thereafter. The agent's objective is to learn an optimal policy π^* that maximizes value at every state. Note that the assumption that the optimal policy is Markov—that it maps from states to actions—is only appropriate if the decision process itself is Markov; almost all RL algorithms simply assume this to be the case.

2.2 State Abstraction

To support decision making when X is too high-dimensional for tractable learning, we turn to state abstraction. Our objective is to find an abstraction function $\phi : X \to Z$ mapping each ground state x to an abstract state representation $z = \phi(x)$, with the hope that learning is tractable in Z (see Figure 1a). We refer to X as *ground states* (and M as the ground MDP), to reflect that they are grounded in the true environment. We interchangeably refer to $z \in Z$ as an abstract state or an abstract representation.

An abstraction $\phi : X \to Z$, when applied to an MDP M, induces a new abstract decision process $M_\phi = (Z, A, T^\pi_{\phi,t}, R^\pi_{\phi,t}, \gamma)$, whose dynamics may depend the current timestep t or the agent's behavior policy π, and, crucially, which is not guaranteed to be Markov.[2]

[2]For an example of a non-Markov abstract decision process, see Appendix C.

Definition 2 (Markov Abstraction). *Given an MDP $M = (X, A, R, T, \gamma)$, initial state distribution P_0, and behavior policy π, a state abstraction $\phi : X \to Z$ is* Markov *if and only if it induces an abstract decision process $M_\phi = (Z, A, R^\pi_{\phi,t}, T^\pi_{\phi,t}, \gamma)$ such that the following holds, for all $a \in A$, $z \in Z$, $k \geq 1$:*

$$\begin{aligned} \textit{Markov}(\phi) : \iff \quad & T^{\pi(k)}_{\phi,t}\left(z_{t+1} | \{a_{t-i}, z_{t-i}\}_{i=0}^{k}\right) = T^{\pi}_{\phi,t}(z_{t+1} | a_t, z_t) \\ \cap \quad & R^{\pi(k)}_{\phi,t}\left(z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^{k}\right) = R^{\pi}_{\phi,t}(z_{t+1}, a_t, z_t). \end{aligned} \tag{2}$$

When ϕ is Markov, we equivalently say that M_ϕ is an MDP, or that Z is a Markov state representation.

One common approach to learning an abstract MDP is to use state aggregation (Li et al., 2006) to group states into abstract states using a fixed weighting scheme $w(x)$, representing how much each ground state x contributes to its abstract state z. In this formulation, the abstract decision process is Markov by construction, and T_ϕ and R_ϕ no longer depend on the policy or timestep. Unfortunately, if the agent changes its policy during learning, the abstract MDP may no longer match the behavior of the original MDP (Abel et al., 2018). Abel et al. show that this discrepancy can have negative consequences for learning, such as causing algorithms with bounded sample complexity in the ground MDP to make an arbitrarily large number of mistakes in the abstract MDP.

Rather than choosing an abstraction and hoping its related abstract MDP reflects the behavior of the ground MDP, we start by defining the abstract decision process to explicitly match that behavior, and then select a compatible abstraction that causes the abstract decision process to be Markov.

3 Related Work

The immediately obvious approach to learning a Markov state representation is to jointly train an abstraction ϕ and a forward model $\widehat{T}(\phi(x), a) \approx \phi(x')$ to predict abstract states. Unfortunately, since ϕ is on both sides of the equation, this approach can result in a trivial abstraction like $\phi(x) \mapsto 0$ for all x, which makes learning $\widehat{T}$ easy, but contains no useful information for decision making. This problem can in theory be mitigated by jointly learning a reward model (Gelada et al., 2019), and in principle that is sufficient; however, such an approach will fail in practice if rewards are sparse or the abstraction must be learned without knowledge of the reward function.

Pixel prediction (Kaiser et al., 2020) is one way to avoid learning a trivial abstraction. However, in stochastic domains, this comes with the challenging task of density estimation over the ground state space, and as a result, performance is about on-par with end-to-end deep RL (Van Hasselt et al., 2019). Moreover, both pixel prediction and the related task of pixel reconstruction (Higgins et al., 2017) are misaligned with the fundamental goal of state abstraction. These approaches train models to perfectly reproduce the relevant ground state, ergo the abstract state must effectively throw away no information. By contrast, the objective of state abstraction is to throw away *as much information as possible*, while preserving what is necessary for decision making.

As an alternative to (or in addition to) learning a forward model, it is sometimes beneficial learn an inverse model. An inverse model $I(a|x', x)$ predicts the action distribution that resulted in a transition between a pair of abstract states. Christiano et al. (2016) studied using an inverse model to learn representations that improve transfer from simulation to real-world problems. Agrawal et al. (2016) jointly estimated forward and inverse dynamics models for robot motion planning, and Pathak et al. (2017) also combined forward and inverse models to create an intrinsic reward to aid exploration.

Since estimating high-dimensional distributions can be challenging, a compelling alternative is contrastive learning (Gutmann & Hyvärinen, 2010), which aims to predict whether a particular state, or sequence of states, came from the distribution in question or some other distribution. Shelhamer et al. (2016) empirically evaluated inverse model estimation and a form of multi-step contrastive learning, as well as other objectives such as pixel reconstruction and jointly training with RL. Van den Oord et al. (2018) applied a temporal version of this technique to learn representations by distinguishing legitimate state transitions from manufactured ones while concurrently training an RL agent. Misra et al. (2019) employed a temporal contrastive loss to learn representations while simultaneously learning to explore the domain efficiently using those representations. Srinivas et al. (2020) combined data augmentation and non-temporal contrastive learning with single states, and recently, Stooke et al. (2020) followed up on this work with a temporal version that performed better.

While many of these methods learn representations that lead to substantial improvements in learning performance, none has explicitly addressed the question of how to learn compact abstract state representations that actually preserve the Markov property (apart from pixel prediction, which is subject to problems we have already discussed). We are the first, to our knowledge, to show that the combination of inverse model estimation and temporal contrastive learning, along with constraints on the agent's policy, is sufficient to learn a Markov state representation.

4 Markov Abstract State Representations

Recall that for a state representation to be Markov (whether ground or abstract), it must be a sufficient statistic for predicting the next state and expected reward, for any action the agent might select. The ground state representation of an MDP is Markov by definition, but abstract states have no such guarantees. To design a loss function that results in a Markov abstraction, we will leverage Definition 2, which means we must first define the abstract transition model $T^{\pi}_{\phi,t}$ and reward function $R^{\pi}_{\phi,t}$, as well as their k-step counterparts.

4.1 Constructing the Abstract Decision Process

For the abstract decision process to reflect the behavior of the ground MDP, it must be defined in terms of ground MDP quantities. We can rewrite the transition probabilities as follows:[3]

$$\Pr(z'|a,z) = \sum_{x' \in z'} \sum_{x \in z} \Pr(x'|a,z,x) \Pr(x|a,z) = \sum_{x' \in z'} \sum_{x \in z} \Pr(x'|a,z,x) \frac{\Pr(a|x)\Pr(x|z)}{\sum_{\tilde{x} \in X} \Pr(a|\tilde{x})\Pr(\tilde{x}|z)},$$

where we use the shorthand $x \in z$ to denote $x \in X : \phi(x) = z$. Thus, for a given behavior policy π, the abstract transition probabilities are:

$$T^{\pi}_{\phi,t}(z'|a,z) := \sum_{x' \in z'} \sum_{x \in z} T(x'|a,x) \frac{\pi_t(a|x) B^{\pi}_{\phi,t}(x|z)}{\sum_{\tilde{x} \in X} \pi_t(a|\tilde{x}) B^{\pi}_{\phi,t}(\tilde{x}|z)}, \tag{3}$$

where $B^{\pi}_{\phi,t}$ is a belief distribution representing the likelihood of each ground state in a particular abstract state (see Appendix A for expanded derivations). We use the same approach to define the abstract reward function:

$$R^{\pi}_{\phi,t}(z',a,z) := \sum_{x' \in z'} \sum_{x \in z} R(x',a,x) \frac{T(x'|a,x)\pi_t(a|x) B^{\pi}_{\phi,t}(x|z)}{T^{\pi}_{\phi,t}(z'|a,z) \sum_{\tilde{x} \in X} \pi_t(a|\tilde{x}) B^{\pi}_{\phi,t}(\tilde{x}|z)}. \tag{4}$$

All that remains is to define the belief distribution $B^{\pi}_{\phi,t}$, which we will do step-by-step.[4] Starting from an arbitrary initial state distribution $P_0(x)$, the agent's (possibly non-stationary) behavior policy $\pi := \{\pi_0, \pi_1, \dots\}$ induces a distribution P^{π}_t over ground states at each subsequent time step:

$$P^{\pi}_t(x) := \sum_{a \in A} \sum_{\tilde{x} \in X} T(x|a,\tilde{x}) \pi_{t-1}(a|\tilde{x}) P^{\pi}_{t-1}(\tilde{x}), \quad \text{for all } t \geq 1. \tag{5}$$

We use $P^{\pi}_t(x)$ to define the belief distribution $B^{\pi}_{\phi,t}(x|z)$, noting that $\Pr(x|z) = \frac{\Pr(z|x)\Pr(x)}{\sum_{\tilde{x} \in X} \Pr(z|\tilde{x})\Pr(\tilde{x})}$:

$$B^{\pi}_{\phi,t}(x|z) := \frac{\mathbb{1}[\phi(x)=z]\ P^{\pi}_t(x)}{\sum_{\tilde{x} \in z} P^{\pi}_t(\tilde{x})}, \tag{6}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. Note that P^{π}_t and $B^{\pi}_{\phi,t}$ may be non-stationary, even if π is a stationary policy.[5]

If we restrict ourselves to a particular class of policies, the abstract transition and reward can be written much more simply. In particular, consider the class of policies Π_{ϕ} where $\pi \in \Pi_{\phi}$ if and only

[3] This section closely follows Hutter (2016), except here we consider distributions over ground states, rather than full histories.

[4] Note that while we use $B^{\pi}_{\phi,t}$ for our theoretical results, the agent need not estimate or maintain a belief distribution in order to learn a Markov abstraction.

[5] This can happen, for example, when the policy induces either a Markov chain that does not have a stationary distribution, or one whose stationary distribution is different from P_0.

if $\pi(\cdot|x) = \pi(\cdot|\tilde{x})$, for all $x, \tilde{x}$ such that $\phi(x) = \phi(\tilde{x})$—in essence, policies defined only in terms of the abstract state z. Whenever $\pi \in \Pi_\phi$, it is fixed for all $x \in z$, and we have:

$$T^\pi_{\phi,t}(z'|a,z) = \sum_{x' \in z'} \sum_{x \in z} T(x'|a,x) B^\pi_{\phi,t}(x|z),$$

$$R^\pi_{\phi,t}(z'|a,z) = \sum_{x' \in z'} \sum_{x \in z} R(x',a,x) \frac{T(x'|a,x) B^\pi_{\phi,t}(x|z)}{T^\pi_{\phi,t}(z'|a,z)}.$$

In the special case where the belief distribution is stationary and policy-independent (and if we interpret rewards as being defined over state-action pairs), $B^\pi_{\phi,t}(x|z)$ is equivalent to the fixed weighting function $w(x)$ of Li et al. (2006).

4.2 Conditioning on Multiple Steps

Definition 2 compares $T^\pi_{\phi,t}$ and $R^\pi_{\phi,t}$ defined above with versions conditioned on k steps of additional history. We can easily modify (3) and (4) to condition on additional history information by generalizing $B^\pi_{\phi,t}$ to a k-step belief distribution $B^{\pi(k)}_{\phi,t}$:

$$B^{\pi(k)}_{\phi,t}(x_t|z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^k) := \frac{\mathbb{1}[\phi(x_t)=z_t] \sum_{x_{t-1} \in X} T(x_t|a_{t-1},x_{t-1})\, B^{\pi(k-1)}_{\phi,t}\left(x_{t-1}|z_{t-1},\{a_{t-i},z_{t-i}\}_{i=2}^k\right)}{\sum_{\tilde{x}_t \in z_t} \sum_{\tilde{x}_{t-1} \in z_{t-1}} T(\tilde{x}_t|a_{t-1},\tilde{x}_{t-1})\, B^{\pi(k-1)}_{\phi,t}\left(\tilde{x}_{t-1}|z_{t-1},\{a_{t-i},z_{t-i}\}_{i=2}^k\right)}, \tag{7}$$

for $k \geq 1$. When $k = 0$ we define $B^{\pi(k)}_{\phi,t} := B^\pi_{\phi,t}$. This definition follows from combining (5) and (6) and conditioning each quantity on the additional history.

Using $B^{\pi(k)}_{\phi,t}$, we define the k-step abstract transition model $T^{\pi(k)}_{\phi,t}$ and reward function $R^{\pi(k)}_{\phi,t}$:

$$T^{\pi(k)}_{\phi,t}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^k) := \sum_{x_{t+1} \in z_{t+1}} \sum_{x_t \in z_t} \frac{T(x_{t+1}|a_t,x_t)\, \pi_t(a_t|x_t) B^{\pi(k)}_{\phi,t}\left(x_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^k\right)}{\sum_{\tilde{x}_t \in X} \pi_t(a_t|\tilde{x}_t) B^{\pi(k)}_{\phi,t}\left(\tilde{x}_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^k\right)}, \tag{8}$$

$$R^{\pi(k)}_{\phi,t}(z_{t+1}, \{a_{t-i},z_{t-i}\}_{i=0}^k) := \sum_{x_{t+1} \in z_{t+1}} \sum_{x_t \in z_t} \frac{R(x_{t+1},a_t,x_t)\, T(x_{t+1}|a_t,x_t)\, \pi_t(a_t|x_t) B^{\pi(k)}_{\phi,t}\left(x_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^k\right)}{T^{\pi(k)}_{\phi,t}\left(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^k\right) \sum_{\tilde{x}_t \in X} \pi_t(a_t|\tilde{x}_t) B^{\pi(k)}_{\phi,t}\left(\tilde{x}_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^k\right)}. \tag{9}$$

Now that we have these extended-history versions of $T^\pi_{\phi,t}$ and $R^\pi_{\phi,t}$, we are ready to apply Definition 2 and describe sufficient conditions for ensuring (2).

4.3 Sufficient Conditions for a Markov Abstraction

The strictly necessary conditions for ensuring an abstraction ϕ is Markov, by Definition 2, depend on T and R, which are typically unknown and hard to estimate due to X's high-dimensionality. However, we can still find conditions that imply Definition 2 without explicitly knowing T and R. To do this, we will require that three quantities are equivalent in M and M_ϕ: the agent's policy, the inverse dynamics model, and a density ratio that we define below.

The inverse dynamics model $I^\pi_t(a|x',x)$ is related to the transition model $T(x'|a,x)$ and the expected next-state model $N^\pi_t(x'|x)$ via Bayes' theorem: $T(x'|a,x) = \frac{I^\pi_t(a|x',x) N^\pi_t(x'|x)}{\pi_t(a|x)}$, where I^π_t and N^π_t are defined as follows:

$$I^\pi_t(a|x',x) := \frac{T(x'|a,x)\pi_t(a|x)}{N^\pi_t(x'|x)} \qquad N^\pi_t(x'|x) := \textstyle\sum_{\tilde{a} \in A} T(x'|\tilde{a},x)\pi_t(\tilde{a}|x).$$

To relate the above quantities to their abstract counterparts in M_ϕ, we also define the following expressions:

$$P^\pi_{\phi,t}(z) := \textstyle\sum_{\tilde{x} \in z} P^\pi_t(\tilde{x}) \qquad \pi_{\phi,t}(a_t|z_t) := \textstyle\sum_{\tilde{x} \in X} \pi_t(a_t|\tilde{x}_t) B^\pi_{\phi,t}(\tilde{x}_t|z_t)$$

$$I^\pi_{\phi,t}(a_t|z_{t+1},z_t) := \frac{T_\phi(z_{t+1}|a_t,z_t)\pi_{\phi,t}(a_t|z_t)}{N^\pi_{\phi,t}(z_{t+1}|z_t)} \qquad N^\pi_{\phi,t}(z_{t+1}|z_t) := \textstyle\sum_{\tilde{a}_t \in A} T_\phi(z_{t+1}|\tilde{a}_t,z_t)\pi_{\phi,t}(\tilde{a}_t|z_t).$$

With these definitions in place, we now describe three conditions in Theorem 1 that together are sufficient for ensuring M_ϕ is an MDP.

Theorem 1. *Let $\phi : X \to Z$ be an abstraction of MDP $M = (X, A, T, R, \gamma)$, with arbitrary initial ground-state distribution P_0, and let π be the agent's behavior policy. Construct the abstract decision process $M_\phi = (Z, A, T^\pi_{\phi,t}, R^\pi_{\phi,t}, \gamma)$, where $T^\pi_{\phi,t}$ is defined via* (3) *and $R^\pi_{\phi,t}$ via* (4)*.*

If all of the following conditions hold at every timestep t:

1. ***Matching Policies.*** *The ground-state policy is piecewise-constant with respect to ϕ:*

$$\begin{aligned} & \phi(x) = \phi(\tilde{x}) \implies \pi_t(\cdot|x) = \pi_t(\cdot|\tilde{x}) && \forall\, x, \tilde{x} \in X \\ \equiv\ & \pi_{\phi,t}(\cdot|z) = \pi_t(\cdot|x) && \forall\, z \in Z, x \in X : \phi(x) = z. \end{aligned} \tag{10a}$$

2. ***Matching Inverse Models.*** *The ground- and abstract-state inverse models are equal:*

$$I^\pi_{\phi,t}(\cdot|z', z) = I^\pi_t(\cdot|x', x) \qquad \forall\, z', z \in Z,\ \forall\, x', x \in X : \phi(x') = z' \cap \phi(x) = z. \tag{10b}$$

3. ***Matching Next-State Density Ratios.*** *The next-abstract-state density ratio is equal to the expectation with respect to the belief distribution of the next-ground-state density ratio:*

$$\frac{N^\pi_{\phi,t}(z'|z)}{P^\pi_{\phi,t}(z')} = \sum_{\tilde{x} \in X} \frac{N^\pi_t(x'|\tilde{x})}{P^\pi_t(x')} B^\pi_{\phi,t}(\tilde{x}|z) \qquad \forall\, z', z \in Z,\ \forall\, x' \in X : \phi(x') = z'. \tag{10c}$$

Then, M_ϕ is Markov as defined in Definition 2.

The proof makes use of two lemmas: Lemma 4.1, that equal k-step and $(k-1)$-step belief distributions imply equal k-step and $(k-1)$-step transition models and reward functions, and Lemma 4.2, that equal k-step and $(k-1)$-step belief distributions imply equal $(k+1)$-step and k-step belief distributions.

We provide a sketch of each proof below, and defer the full proofs to Appendix B.

Lemma 4.1. *Given an MDP M, abstraction ϕ, policy π, initial state distribution P_0, and any $k \geq 1$, if it holds that $B^{\pi(k)}_{\phi,t}(x_t|z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^k) = B^{\pi(k-1)}_{\phi,t}(x_t|z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^{k-1})$,*

Then for all $a_t \in A$ and $z_{t+1} \in Z$,

$$\begin{aligned} & T^{\pi(k)}_{\phi,t}(z_{t+1}|\{a_{t-i}, z_{t-i}\}_{i=0}^k) = T^{\pi(k-1)}_{\phi,t}(z_{t+1}|\{a_{t-i}, z_{t-i}\}_{i=0}^{k-1}) \\ \cap \quad & R^{\pi(k)}_{\phi,t}(z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^k) = R^{\pi(k-1)}_{\phi,t}(z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^{k-1}). \end{aligned}$$

Proof sketch: Starting with $B^{\pi(k)}_{\phi,t} = B^{\pi(k-1)}_{\phi,t}$, repeatedly multiply and divide both sides by the same quantities, or take the same summations of both sides, to obtain $T^{\pi(k)}_{\phi,t} = T^{\pi(k-1)}_{\phi,t}$. Then apply the same process again, making use of the fact that $T^{\pi(k)}_{\phi,t} = T^{\pi(k-1)}_{\phi,t}$, to obtain $R^{\pi(k)}_{\phi,t} = R^{\pi(k-1)}_{\phi,t}$. □

Lemma 4.2. *Given an MDP M, abstraction ϕ, policy π, and initial state distribution P_0, if for all $t \geq k, z_t \in Z, x_t \in X : \phi(x_t) = z_t$, we have that $B^{\pi(k)}_{\phi,t}(x_t|z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^k) = B^{\pi(k-1)}_{\phi,t}(x_t|z_t, \{a_{t-i}, z_{t-i}\}_{i=1}^{k-1})$, then for all $z_{t+1} \in Z, x_{t+1} \in X : \phi(x_{t+1}) = z_{t+1}$,*

$$B^{P^\pi_t}_{\phi,k+1}\left(x_{t+1}\middle|z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^k\right) = B^{\pi(k)}_{\phi,t}\left(x_{t+1}\middle|z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^{k-1}\right)$$

Proof sketch: Let $T^{\pi(k)}_{\phi,t}$ be defined via (8). Then by Lemma 4.1, $T^{\pi(k)}_{\phi,t} = T^{\pi(k-1)}_{\phi,t}$. Use this fact while following the same approach as in Lemma 4.1. □

Now we summarize the proof of the main theorem.

Proof of Theorem 1 (sketch): The structure of the proof is as follows. We begin by showing that the belief distributions $B^{\pi(k)}_{\phi,t}$ and $B^{\pi(k-1)}_{\phi,t}$ must be equal for $k = 1$, and use Lemma 4.1 to prove the base case of the theorem. Then we use Lemmas 4.2 and 4.1 together, along with induction on k to prove that the theorem holds in general. □

Theorem 1 describes a set of sufficient conditions under which ϕ results in a Markov state representation. Of course, we have no way during learning of directly ensuring these conditions are true. In the next section we define a representation learning objective based on approximately satisfying these conditions that is differentiable and suitable for learning ϕ using deep neural networks.

5 Training a Markov Abstraction

Matching policies. To ensure that the policies match, we restrict the policy class to Π_ϕ by defining π as a mapping from $Z \to \Pr(A)$, rather than from $X \to \Pr(A)$.

Matching inverse models. To ensure the ground-state and abstract-state inverse models match, we consider a batch of experiences (x_i, a_i, x'_i), encode ground states with ϕ, and jointly train a model $f(a|\phi(x'), \phi(x); \theta)$ to predict a distribution over actions, with a_i as the label. This can be achieved by minimizing cross-entropy, whether the action space is discrete or continuous:[6]

$$\mathcal{L}_{inverse} := -\frac{1}{N} \sum_{i=1} \log f(a = a_i | \phi(x'_i), \phi(x_i); \theta)$$

Matching next-state density ratios. The third objective, that $\frac{N^\pi_{\phi,t}(z'|z)}{P^\pi_{\phi,t}(z')} = \mathbb{E}_{B^\pi_{\phi,t}(x|z)} \left[\frac{N^\pi_t(x'|x)}{P^\pi_t(x')} \right]$, says we can distinguish conditional samples from marginal samples equally well for abstract states or ground states. We can encode this objective as a type of contrastive loss.

Suppose we have a dataset consisting of samples $X_c = \{{x'_c}^{(i)}\}_{i=1}^{n_c}$ drawn from the conditional distribution $\Pr(x'|x)$, and samples $X_m = \{{x'_m}^{(j)}\}_{j=1}^{n_m}$ drawn from the marginal $\Pr(x')$. We assign label $y = 1$ to samples from X_c and $y = 0$ to samples from X_m. We can define alternative names for the two distributions $p(x'|y = 1) := \Pr(x'|x)$ and $p(x'|y = 0) := \Pr(x')$. We are concerned with the density ratio $\delta(x') := \frac{\Pr(x'|x)}{\Pr(x')}$, which we rewrite according to the derivation of Tiao (2017):

$$\delta(x') = \frac{p(x'|y=1)}{p(x'|y=0)} = \frac{p(y=1|x')p(x')}{p(y=1)} \frac{p(y=0)}{p(y=0|x')p(x')} = \frac{n_m}{(n_m+n_c)} \frac{(n_m+n_c)}{n_c} \frac{p(y=1|x')}{p(y=0|x')} = \frac{n_m}{n_c} \frac{p(y=1|x')}{1-p(y=1|x')}. \tag{11}$$

Similarly, for abstract states $z' = \phi(x')$, we have $\delta_\phi(z') := \frac{\Pr(z'|z)}{\Pr(z')} = \frac{n_m}{n_c} \frac{q(y=1|z')}{1-q(y=1|z')}$, where q is our renamed distribution.

We train a function approximator $g(x', x; \psi)$ to estimate $p(y = 1|x')$. If we wanted to estimate $\delta(x')$, we could then substitute g for $p(y = 1|x')$ in (11); however, to satisfy condition (10c), we only need to ensure $\delta(x') = \delta_\phi(z')$. We achieve this by modifying our estimator so it is trained jointly with the abstraction function ϕ, minimizing the cross-entropy between the predictions $g(y|\phi(x'_i), \phi(x_i); \psi)$ and labels y_i:

$$\mathcal{L}_{contrastive} := -\frac{1}{n_c + n_m} \sum_{i=1}^{n_c+n_m} \log g(y = y_i | \phi(x'_i), \phi(x_i); \psi).$$

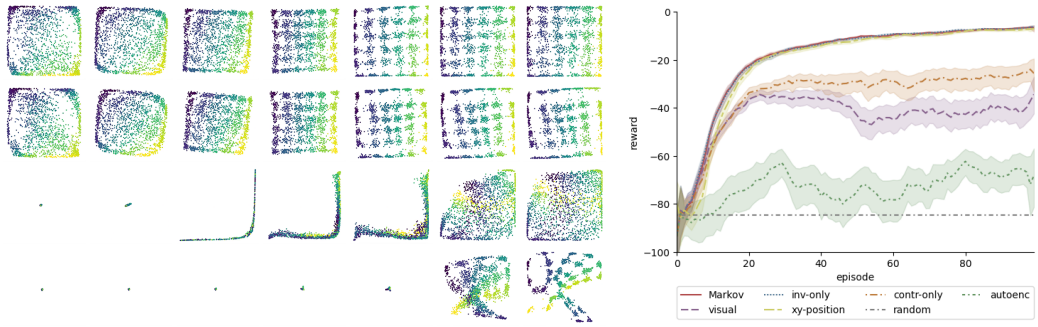
In doing so, we ensure that g approaches q and p simultaneously, which drives $\delta_\phi(z') \to \delta(x')$.

Markov abstraction loss. We generate a batch of experiences using a uniform random policy, which is a member of Π_ϕ, then minimize a weighted combination of the inverse model and contrastive losses above, while training ϕ end-to-end:

$$\arg\min_{\phi,\theta,\psi} \mathcal{L}_{Markov} := \alpha \mathcal{L}_{inverse} + \beta \mathcal{L}_{contrastive}.$$

The inverse and contrastive losses avoid the common pitfall of learning a trivial abstraction like $\phi(x) \mapsto 0$, without requiring any reward information. While such an abstraction would technically satisfy the ratio condition (10c), it does not minimize $\mathcal{L}_{Markov}$, since $\phi(x)$ contains no useful information for predicting actions or distinguishing $x' \sim \Pr(x'|x)$ from $x' \sim \Pr(x')$. This partially explains why inverse models and temporal contrastive losses are so effective for representation learning (Shelhamer et al., 2016; Pathak et al., 2017; Van den Oord et al., 2018; Anand et al., 2019; Misra et al., 2019; Stooke et al., 2020). Note that when the policy matching condition (10a) holds, and the policy is stationary and deterministic, then $I^\pi_{\phi,t}(a|z', z) = \pi_\phi(z) = \pi(x) = I^\pi_t(a|x', x)$ and the inverse matching condition is satisfied trivially. Thus we expect $\mathcal{L}_{inverse}$ to be most useful for representation learning when the policy has high entropy or is rapidly changing.

[6] For continuous actions, we can alternatively assume deterministic policies and train via regression.



(a) (b)

Figure 3: (a) Visualization of the learning progress at selected times (left to right) of a 2-D Markov state abstraction for the 6×6 visual gridworld domain. (Top row) $\mathcal{L}_{Markov}$; (second row) $\mathcal{L}_{inverse}$ only; (third row) $\mathcal{L}_{contrastive}$ only; (last row) autoencoder. Color denotes ground-truth (x, y) position, which is not shown to the agent. (b) Mean episode reward for the visual gridworld navigation task (300 seeds; 5-point moving average; shaded regions denote 95% confidence intervals).

6 Proof-of-Concept Experiment

We test our approach on a visual gridworld domain.[7] Each discrete (x, y) position in the 6×6 gridworld is mapped to a noisy image like the one in Figure 1a. We emphasize that agent only sees these images; it does not have access to the ground-truth (x, y) position.

First we gather a batch of experiences using a uniform random exploration policy in a version of the gridworld with no rewards or terminal states. We train an abstraction function ϕ_{Markov} by minimizing $\mathcal{L}_{Markov}$ (with $\alpha = \beta = 1$), and we visualize the latent space in Figure 2a (top row). We compare against ablations that train with only $\mathcal{L}_{inverse}$ or $\mathcal{L}_{contrastive}$, as well as against a baseline autoencoder that we train via pixel reconstruction.

We see that ϕ_{Markov} and $\phi_{inverse}$ cluster the noisy observations and recover the 6×6 grid structure, whereas the other two losses do not generally have an obvious interpretation. We found that $\phi_{inverse}$ usually converged to a representation that separated the edges and corner states from the central states, since the self-loop transitions at edge and corner states led to different action distributions. We also observed that $\phi_{contrastive}$ and $\phi_{autoencoder}$ frequently failed to converge at all (see Appendix F for more visualizations).

Next we froze the representation and trained a DQN (Mnih et al., 2015), using ϕ_{Markov} to map images to abstract states. We compared performance against $\phi_{inverse}$, $\phi_{contrastive}$, and $\phi_{autoencoder}$, as well as end-to-end DQN with no pretraining. We also compared against a uniform random policy and DQN trained on ground-truth (x, y) position (with no abstraction). We plot learning curves in Fig. 2b for 300 random seeds each. Markov abstractions match the performance of ground-truth position, and beat every other learned representation except $\phi_{inverse}$, which performs similarly, although, by itself, $\mathcal{L}_{inverse}$ does not always produce Markov abstractions (see counterexample in Appendix C).

7 Conclusion

We have demonstrated a principled approach to learning abstract state representations that provably results in Markov abstract states. Prior work on representation learning showed promising results, but lacked this important theoretical consideration. We defined what it means for an abstract state representation to be Markov while ensuring that the abstract MDP faithfully reflects the dynamics of its ground MDP. We then described a set of sufficient conditions for achieving such an abstract MDP. We adapted these conditions into a practical training objective that combines inverse model estimation and temporal contrastive learning, along with policy constraints. Our approach learns a state representation that captures the underlying structure of the domain and closes the performance gap between learning with visual features and using ground-truth non-visual state. In future work, we plan to leverage this technique to learn representations for more challenging domains.

[7] Code and videos are available at https://github.com/camall3n/markov-state-abstractions

Acknowledgments and Disclosure of Funding

We thank Ben Abbatematteo, David Abel, Barrett Ames, Kavosh Asadi, Akhil Bagaria, Alexander Ivanov, Michael Littman, Sam Lobel, and our other colleagues at Brown for their thoughtful advice and countless helpful discussions. This research was supported by the ONR under the PERISCOPE MURI Contract N00014-17-1-2699.

References

Abel, D., Arumugam, D., Lehnert, L., and Littman, M. L. State abstractions for lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018.

Agrawal, P., Nair, A., Abbeel, P., Malik, J., and Levine, S. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R. D. Unsupervised state representation learning in Atari. In *Advances in Neural Information Processing Systems*, pp. 8769–8782, 2019.

Christiano, P., Shah, Z., Mordatch, I., Schneider, J., Blackwell, T., Tobin, J., Abbeel, P., and Zaremba, W. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv*, preprint 1610.03518, 2016.

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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1665–1674, 2019.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. In *IEEE International Conference on Robotics and Automation*, pp. 512–519, 2016.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., and Bellemare, M. G. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2170–2179, 2019.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Ha, D. and Schmidhuber, J. World models. *arXiv*, preprint 1803.10122, 2018.

Van Hasselt, H., Hessel, M., and Aslanides, J. When to use parametric models in reinforcement learning? In *Advances in Neural Information Processing Systems*, pp. 14322–14333, 2019.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. DARLA: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1480–1490, 2017.

Hutter, M. Extreme state aggregation beyond Markov decision processes. *Theoretical Computer Science*, 650:73–91, 2016.

Kaiser, Ł., Babaeizadeh, M., Miłos, P., Osiński, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model based reinforcement learning for Atari. In *International Conference on Learning Representations*, 2020.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *arXiv*, preprint 2004.14990, 2020.

Li, L., Walsh, T. J., and Littman, M. L. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, pp. 531–539, 2006.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. *arXiv*, preprint 1911.05815, 2019.

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*, 518(7540):529–533, 2015.

Van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv*, preprint 1807.03748, 2018.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 488–489, 2017.

Shelhamer, E., Mahmoudieh, P., Argus, M., and Darrell, T. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv*, preprint 1612.07307, 2016.

Song, Z., Parr, R., Liao, X., and Carin, L. Linear feature encoding for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4224–4232, 2016.

Srinivas, A., Laskin, M., and Abbeel, P. CURL: Contrastive unsupervised representations for reinforcement learning. *arXiv*, preprint 2004.04136, 2020.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. *arXiv*, preprint 2009.08319, 2020.

Tiao, L. A simple illustration of density ratio estimation and KL divergence estimation by probabilistic classification. http://louistiao.me/notes/a-simple-illustration-of-density-ratio-estimation-and-kl-divergence-estimation-by-probabilistic-classification, 2017. [Online; accessed 16-March-2020].

Zhang, A., Satija, H., and Pineau, J. Decoupling dynamics and reward for transfer learning. *arXiv*, preprint 1804.10689, 2018.

A Expanded Derivations

A.1 Abstract transition probabilities

$$
\begin{aligned}
&\Pr(z'|a,z) \\
&= \sum_{x' \in X} \Pr(z'|x',a,z) \Pr(x'|a,z) \\
&= \sum_{x' \in X:\phi(x')=z'} \Pr(x'|a,z) \\
&= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X} \Pr(x'|a,x,z) \Pr(x|a,z) \\
&= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X} \Pr(x'|a,x,z) \frac{\Pr(a|x,z) \Pr(x|z)}{\sum_{\tilde{x} \in X} \Pr(a|\tilde{x},z) \Pr(\tilde{x}_t|z_t)} \\
T^{\pi}_{\phi,t}(z'|a,z) &:= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X:\phi(x)=z} T(x'|a,x) \frac{\pi_t(a|x) B^{\pi}_{\phi,t}(x|z)}{\sum_{\tilde{x} \in X} \pi_t(a|\tilde{x}) B^{\pi}_{\phi,t}(\tilde{x}|z)}
\end{aligned}
$$

A.2 Abstract rewards

$$
\begin{aligned}
&\sum_{r \in R} r \Pr(r|z',a,z) \\
&= \sum_{x' \in X} \sum_{x \in X} \sum_{r \in R} r \Pr(r,x',x|z',a,z) \\
&= \sum_{x' \in X} \sum_{x \in X} \sum_{r \in R} r \Pr(r|x',z',a,x,z) \Pr(x',x|z',a,z) \\
&= \sum_{x' \in X} \sum_{x \in X} R(x',a,x) \frac{\Pr(z'|x',a,x,z) \Pr(x',x|a,z)}{\Pr(z'|a,z)} \\
&= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X} R(x',a,x) \frac{\Pr(x'|a,x,z) \Pr(x|a,z)}{\Pr(z'|a,z)} \\
&= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X} R(x',a,x) \frac{\Pr(x'|a,x)}{\Pr(z'|a,z)} \frac{\Pr(a|x) \Pr(x|z)}{\sum_{\tilde{x} \in X} \Pr(a|\tilde{x}) \Pr(\tilde{x}|z)} \\
R^{\pi}_{\phi,t}(z',a,z) &:= \sum_{x' \in X:\phi(x')=z'} \sum_{x \in X:\phi(x)=z} R(x',a,x) \frac{T(x'|a,x)\, \pi_t(a|x) B^{\pi}_{\phi,t}(x|z)}{T^{\pi}_{\phi,t}(z'|a,z) \sum_{\tilde{x} \in X} \pi_t(a|\tilde{x}) B^{\pi}_{\phi,t}(\tilde{x}|z)}
\end{aligned}
$$

B Proofs

Proof of Lemma 4.1:

$$B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)=B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)$$

Let $a_t \in A$ be any action.

$$\begin{aligned}
&\Rightarrow \quad \pi_t(a_t|x_t)B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)=\pi_t(a_t|x_t)B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)\\
&\Rightarrow \quad \frac{\pi_t(a_t|x_t)B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)}{\sum_{\tilde{x}_t\in X}\pi_t(a_t|\tilde{x}_t)B_{\phi,t}^{\pi(k)}\left(\tilde{x}_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)}=\frac{\pi_t(a_t|x_t)B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)}{\sum_{\tilde{x}_t\in X}\pi_t(a_t|\tilde{x}_t)B_{\phi,t}^{\pi(k-1)}\left(\tilde{x}_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)}
\end{aligned} \tag{12}$$

Let

$$C_{\phi,t}^{\pi(k)}:=\frac{\pi_t(a_t|x_t)B_{\phi,t}^{\pi(k)}\left(x_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)}{\sum_{\tilde{x}_t\in X}\pi_t(a_t|\tilde{x}_t)B_{\phi,t}^{\pi(k)}\left(\tilde{x}_t|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)} \tag{13}$$

Combining (12) and (13), we obtain:

$$C_{\phi,t}^{\pi(k)}=C_{\phi,t}^{\pi(k-1)} \tag{14}$$

$$\Rightarrow \quad \sum_{x_t\in z_t}\sum_{x_{t+1}\in z_{t+1}}T\left(x_{t+1}\mid a_t,x_t\right)C_{\phi,t}^{\pi(k)}=\sum_{x_t\in z_t}\sum_{x_{t+1}\in z_{t+1}}T\left(x_{t+1}\mid a_t,x_t\right)C_{\phi,t}^{\pi(k-1)}$$

$$\Leftrightarrow \quad T_{\phi,t}^{\pi(k)}\left(z_{t+1}\mid\{a_{t-i},z_{t-i}\}_{i=0}^{k}\right)=T_{\phi,t}^{\pi(k-1)}\left(z_{t+1}\mid\{a_{t-i},z_{t-i}\}_{i=0}^{k-1}\right) \tag{15}$$

Additionally, we can combine (14) and (15) and apply the same approach for rewards:

$$\begin{aligned}
&C_{\phi,t}^{\pi(k)}=C_{\phi,t}^{\pi(k-1)}\\
&\Rightarrow \quad \frac{C_{\phi,t}^{\pi(k)}}{T_{\phi,t}^{\pi(k)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k})}=\frac{C_{\phi,t}^{\pi(k-1)}}{T_{\phi,t}^{\pi(k-1)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k-1})}\\
&\Rightarrow \quad \frac{T(x_{t+1}|a_t,z_t)C_{\phi,t}^{\pi(k)}}{T_{\phi,t}^{\pi(k)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k})}=\frac{T(x_{t+1}|a_t,z_t)C_{\phi,t}^{\pi(k-1)}}{T_{\phi,t}^{\pi(k-1)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k-1})}\\
&\Rightarrow \quad R_{\phi,t}^{\pi(k)}\left(z_{t+1},\{a_t,z_t\}_{i=0}^{k}\right)=R_{\phi,t}^{\pi(k-1)}\left(z_{t+1},\{a_t,z_t\}_{i=0}^{k-1}\right)
\end{aligned} \tag{16}$$

□

Proof of Lemma 4.2:

Let $T_{\phi,t}^{\pi(k)}$ be defined via (8). Then applying Lemma 4.1 to the premise gives:

$$T_{\phi,t}^{\pi(k)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k})=T_{\phi,t}^{\pi(k-1)}(z_{t+1}|\{a_{t-i},z_{t-i}\}_{i=0}^{k-1})$$

Returning to the premise, we have:

$$\begin{aligned}
&B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)=B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)\\
&\Rightarrow \quad \frac{B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)}{T_{\phi,t}^{\pi(k)}\left(z_{t+1}\middle|\{a_{t-i},z_{t-i}\}_{i=0}^{k}\right)}=\frac{B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)}{T_{\phi,t}^{\pi(k-1)}\left(z_{t+1}\middle|\{a_{t-i},z_{t-i}\}_{i=0}^{k-1}\right)}\\
&\Rightarrow \sum_{\substack{x_t\in X:\\ \phi(x_t)=z_t}}\frac{T(x_{t+1}|a_t,x_t)B_{\phi,t}^{\pi(k)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k}\right)}{T_{\phi,t}^{\pi(k)}\left(z_{t+1}\middle|\{a_{t-i},z_{t-i}\}_{i=0}^{k}\right)}=\sum_{\substack{x_t\in X:\\ \phi(x_t)=z_t}}\frac{T(x_{t+1}|a_t,x_t)B_{\phi,t}^{\pi(k-1)}\left(x_t\middle|z_t,\{a_{t-i},z_{t-i}\}_{i=1}^{k-1}\right)}{T_{\phi,t}^{\pi(k-1)}\left(z_{t+1}\middle|\{a_{t-i},z_{t-i}\}_{i=0}^{k-1}\right)}\\
&\Rightarrow \quad B_{\phi,t}^{\pi(k+1)}\left(x_{t+1}\middle|z_{t+1},\{a_{t-i},z_{t-i}\}_{i=0}^{k}\right)=B_{\phi,t}^{\pi(k)}\left(x_{t+1}\middle|z_{t+1},\{a_{t-i},z_{t-i}\}_{i=0}^{k-1}\right)
\end{aligned}$$

□

Proof of Theorem 1:

Base case. Let $B^{\pi}_{\phi,t}$ be defined via (6). Then for any $z_{t+1}, z_t \in Z$, and $x_{t+1} \in X$ such that $\phi(x_{t+1}) = z_{t+1}$, and any action $a_{t-1} \in A$:

$$\begin{aligned}
\frac{N^{\pi}_{\phi,t}(z_t|z_{t-1})}{P^{\pi}_{\phi,t}(z_t)} &= \sum_{x_{t-1}\in X} \frac{N^{\pi}_{t}(x_t|x_{t-1})}{P^{\pi}_{t}(x_t)} B^{\pi}_{\phi,t}(x_{t-1}|z_{t-1}) \\
\Rightarrow \quad \frac{P^{\pi}_{t}(x_t)}{P^{\pi}_{\phi,t}(z_t)} &= \sum_{x_{t-1}\in X} \frac{N^{\pi}_{t}(x_t|x_{t-1}) B^{\pi}_{\phi,t}(x_{t-1}|z_{t-1})}{N^{\pi}_{\phi,t}(z_t|z_{t-1})} \cdot \frac{I^{\pi}_{t}(a_{t-1}|x_t,x_{t-1})\pi_{\phi,t}(a_{t-1}|z_{t-1})}{I^{\pi}_{\phi,t}(a_{t-1}|z_t,z_{t-1})\pi_t(a_{t-1}|x_{t-1})} \\
\Rightarrow \quad \frac{\mathbb{1}[\phi(x_t) = z_t]\, P^{\pi}_{t}(x_t)}{\sum_{\tilde{x}_t \in X} P^{\pi}_{t}(\tilde{x}_t)} &= \mathbb{1}[\phi(x_t) = z_t] \sum_{x_{t-1}\in X} \frac{T(x_t|a_{t-1},x_{t-1}) B^{\pi}_{\phi,t}(x_{t-1}|z_{t-1})}{T^{\pi}_{\phi,t}(z_t|a_{t-1},z_{t-1})} \\
\Rightarrow \quad B^{\pi}_{\phi,t}(x_t|z_t) &= \frac{\mathbb{1}[\phi(x_t) = z_t] \sum_{x_{t-1}\in X} T(x_t|a_{t-1},x_{t-1}) B^{\pi}_{\phi,t}(x_{t-1}|z_{t-1})}{\sum\limits_{\substack{\tilde{x}_t \in z_t \\ \tilde{x}_{t-1}\in z_{t-1}}} \sum T(\tilde{x}_t|a_{t-1},\tilde{x}_{t-1}) \frac{\pi_t(a_{t-1}|\tilde{x}_{t-1}) B^{\pi}_{\phi,t}(\tilde{x}_{t-1}|z_{t-1})}{\sum_{\tilde{\tilde{x}}_{t-1}\in X} \pi_t(a_{t-1}|\tilde{\tilde{x}}_{t-1}) B^{\pi}_{\phi,t}(\tilde{\tilde{x}}_{t-1}|z_{t-1})}} \\
\Rightarrow \quad B^{\pi}_{\phi,t}(x_t|z_t) &= \frac{\mathbb{1}[\phi(x_t) = z_t] \sum_{x_{t-1}\in X} T_{\phi}(x_t|a_{t-1},x_{t-1}) B^{\pi}_{\phi,t}(x_{t-1}|z_{t-1})}{\sum\limits_{\substack{\tilde{x}_t \in z_t \\ \tilde{x}_{t-1}\in z_{t-1}}} \sum T_{\phi}(\tilde{x}_t|a_{t-1},\tilde{x}_{t-1}) B^{\pi}_{\phi,t}(\tilde{x}_{t-1}|z_{t-1})} \\
\Rightarrow \quad B^{\pi(0)}_{\phi,t}(x_t|z_t) &= B^{\pi(1)}_{\phi,t}(x_t|z_t, a_{t-1}, z_{t-1}) \qquad (17)
\end{aligned}$$

Here (17) satisfies the conditions of Lemma 4.1 (with $k = 1$), therefore, for all $a_t \in A$:

$$\begin{aligned}
& T^{\pi(0)}_{\phi,t}(z_{t+1}|a_t, z_t) = T^{\pi(1)}_{\phi,t}(z_{t+1}|a_t, z_t, a_{t-1}, z_{t-1}) \\
\cap \quad & R^{\pi(0)}_{\phi,t}(z_{t+1}, a_t, z_t) = R^{\pi(1)}_{\phi,t}(z_{t+1}, a_t, z_t, a_{t-1}, z_{t-1})
\end{aligned}$$

This proves the theorem for $k = 1$.

Induction on k. Here we note that (17) also satisfies the conditions of Lemma 4.2 (again with $k = 1$), and that Lemma 4.2 holds for arbitrary k. Therefore, by induction on k:

$$B^{\pi(k+1)}_{\phi,t}\left(x_{t+1}\middle|z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^{k}\right) = B^{\pi}_{\phi,t}(x_{t+1}|z_{t+1}) \quad \forall\, k \geq 1 \qquad (18)$$

When (18) holds, we informally say that the belief distribution is Markov. Finally, applying Lemma 4.1 for each subsequent value of $k \geq 1$, we obtain:

$$\begin{aligned}
& T^{\pi(k)}_{\phi,t}\left(z_{t+1}|\{a_{t-i}, z_{t-i}\}_{i=0}^{k}\right) = T^{\pi}_{\phi,t}(z_{t+1}|a_t, z_t) \\
\cap \quad & R^{\pi(k)}_{\phi,t}\left(z_{t+1}, \{a_{t-i}, z_{t-i}\}_{i=0}^{k}\right) = R^{\pi}_{\phi,t}(z_{t+1}, a_t, z_t)
\end{aligned}$$

Therefore, by Definition 2, M_{ϕ} is Markov. □

Corollary 1.1. *Let ϕ, M, P_0, π, M_{ϕ}, $B^{\pi}_{\phi,t}$, and $B^{\pi(k)}_{\phi,t}$ be defined as above. If $B^{\pi(k)}_{\phi,t} = B^{\pi}_{\phi,t}$ for all $k \geq 1$, then ϕ is Markov by Definition 2.*

Proof: The proof follows directly from Lemma 4.1. □

Corollary 1.2. *If there exists some $n \geq 1$ such that $B^{\pi(n)}_{\phi,t} \neq B^{\pi}_{\phi,t}$, then $B^{\pi(1)}_{\phi,t} \neq B^{\pi}_{\phi,t}$.*

Proof: Suppose such an n exists, and assume for the sake of contradiction that $B^{\pi(1)}_{\phi,t} = B^{\pi}_{\phi,t}$. Then by Lemma 4.2, $B^{\pi(k)}_{\phi,t} = B^{\pi}_{\phi,t}$ for all $k \geq 1$. However this is impossible, since we know there exists some $n \geq 1$ such that $B^{\pi(n)}_{\phi,t} \neq B^{\pi}_{\phi,t}$. Therefore $B^{\pi(1)}_{\phi,t} \neq B^{\pi}_{\phi,t}$. □

C Counterexample: Inverse Loss not Sufficient for Markov Abstraction

$$(\mathcal{L}_{inverse} = 0) \implies \text{Markov}(\phi)$$

We provide a counterexample demonstrating that inverse model estimation alone is insufficient to produce a Markov abstraction. In the MDP of Figure 4, there are four numbered states and two actions: "right" and "left". The abstraction ϕ maps ground states 1 and 3 to abstract state A, and ground states 2 and 4 to abstract states B and C, respectively.

Any valid transition between two ground states uniquely identifies the selected action as either "right" or "left". The same is true for their corresponding abstract states. Therefore, the abstraction satisfies the inverse-model matching condition (10b), namely that $I^{\pi}_{\phi,t}(a|z', z) = I^{\pi}_{t}(a|x', x)$ for every $x \in z$ and $x' \in z'$.

However, conditioning the abstract transition model on additional history changes the transition probabilities. For example, suppose P_0 is uniform over ground states 2 and 4, and consider the action sequence ["left", "right"]. When taking action "right" at time $t = 1$, the 1-step probability of transitioning from abstract state A to abstract state B is 0.5. However, given the additional history of having started in B, the probability of transitioning back to B is 1.0, since the extra history collapses the belief distribution at time $t = 1$ to only ground state 1. Since conditioning on additional history changes the abstract transition probabilities, the abstraction is not Markov, by Definition 2.

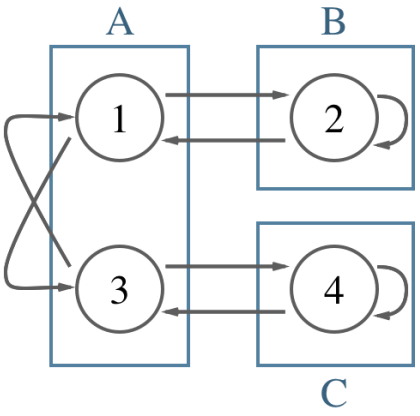


Figure 4: A four-state, two-action MDP and an abstraction function, where ground states are denoted by circles, abstract states are denoted by colored boxes, and actions are denoted by arrows.

D Hyperparameters

Hyperparameter	Value
Number of seeds	300
Optimizer	Adam
Learning rate	0.003
Batch size	2048
Gradient updates	3000
Latent dimensions	2
Number of conditional samples, n_c	1
Number of marginal samples, n_m	1
Loss coefficient, α	1.0
Loss coefficient, β	1.0

Table 1: Hyperparameters for pretraining

Hyperparameter	Value
Number of seeds	300
Number of episodes	100
Maximum steps per episode	100
Optimizer	Adam
Learning rate	0.003
Batch size	16
Discount factor, γ	0.9
Starting exploration probability, ϵ_0	1.0
Final exploration probability, ϵ	0.05
Epsilon decay period	2500
Replay buffer size	10000
Initialization steps	500
Target network copy period	50

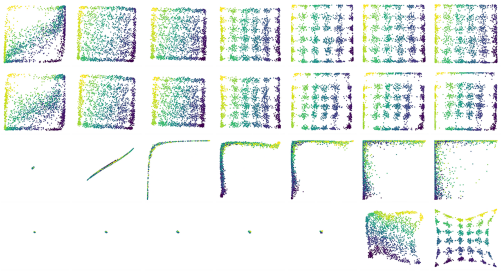
Table 2: Hyperparameters for DQN

E Network Architecture

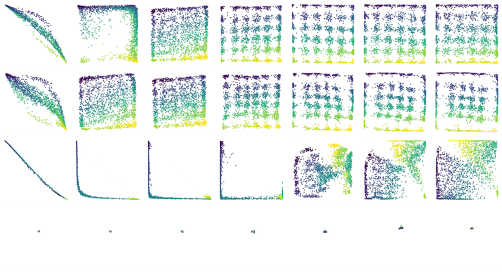
```
FeatureNet(
  (phi): Encoder(
    (phi): Sequential(
      (0): Reshape(-1, 252)
      (1): Linear(in_features=252, out_features=32, bias=True)
      (2): Tanh()
      (3): Linear(in_features=32, out_features=2, bias=True)
      (4): Tanh()
    )
  )
  (inv_model): InverseNet(
    (inv_model): Sequential(
      (0): Linear(in_features=4, out_features=32, bias=True)
      (1): Tanh()
      (2): Linear(in_features=32, out_features=4, bias=True)
    )
  )
  (discriminator): ContrastiveNet(
    (model): Sequential(
      (0): Linear(in_features=4, out_features=32, bias=True)
      (1): Tanh()
      (2): Linear(in_features=32, out_features=1, bias=True)
      (3): Sigmoid()
    )
  )
  (cross_entropy): CrossEntropyLoss()
  (bce_loss): BCELoss()
)

AutoEncoder(
  (phi): Encoder(
    (phi): Sequential(
      (0): Reshape(-1, 252)
      (1): Linear(in_features=252, out_features=32, bias=True)
      (2): Tanh()
      (3): Linear(in_features=32, out_features=2, bias=True)
      (4): Tanh()
    )
  )
  (reverse_phi): Decoder(
    (phi): Sequential(
      (0): Linear(in_features=2, out_features=32, bias=True)
      (1): Tanh()
      (2): Linear(in_features=32, out_features=252, bias=True)
      (3): Tanh()
      (4): Reshape(-1, 21, 12)
    )
  )
  (mse): MSELoss()
)
```

F Additional Representation Visualizations

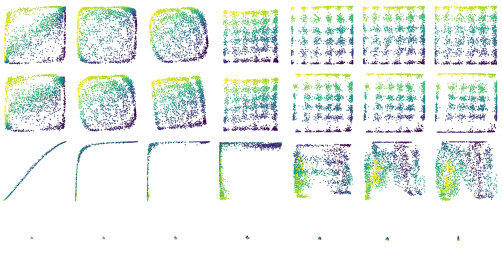
Here we visualize abstraction learning progress for the 6×6 visual gridworld domain across six random seeds. Each figure below displays selected frames (progressing from left to right) of a different abstraction learning method (top to bottom): $\mathcal{L}_{Markov}$; $\mathcal{L}_{inverse}$ only; $\mathcal{L}_{contrastive}$ only; and autoencoder. Color denotes ground-truth (x, y) position, which is not shown to the agent. These visualizations span 30,000 training steps. The third column from the right shows the representations after 3000 steps, which we use for the results in the main text. We show additional learning curves for the final representations in Appendix G.



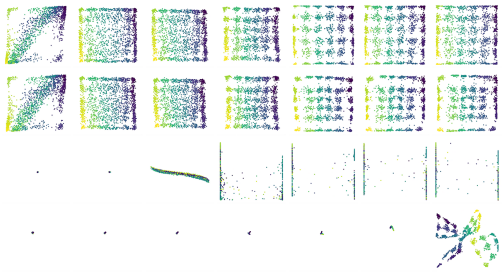
Seed 1



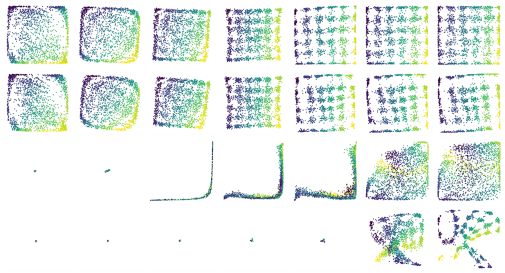
Seed 2



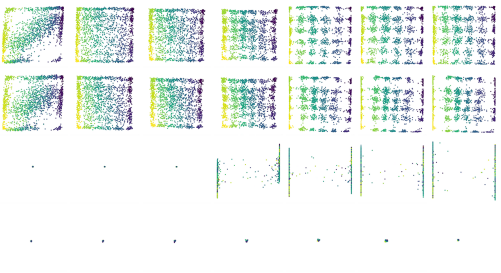
Seed 3



Seed 4



Seed 5



Seed 6

G Increased Pretraining Time

Increasing the number of pretraining steps from 3000 to 30,000 improves the learning performance of $\phi_{contrastive}$ and $\phi_{autoencoder}$, though not enough to dramatically change the results (see Figure 6).

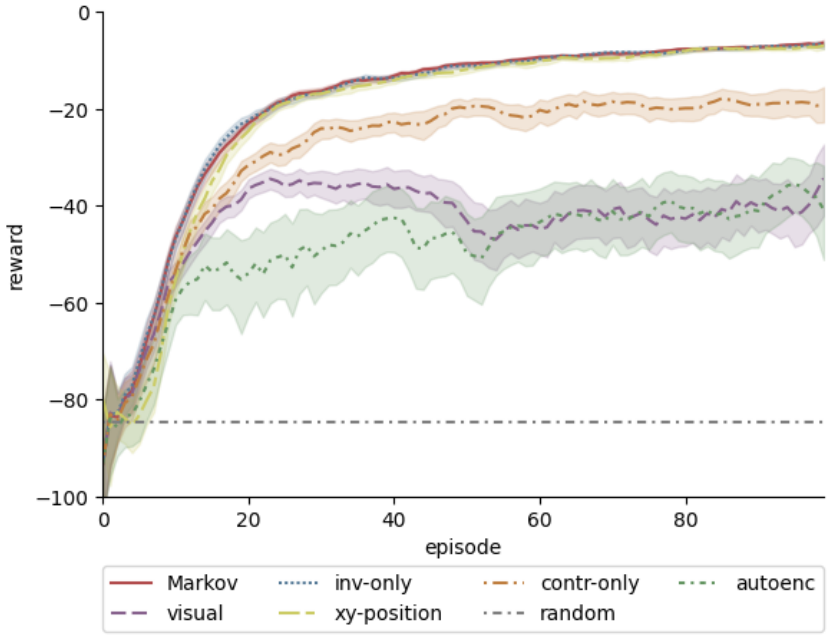


Figure 6: Mean episode reward for the visual gridworld navigation task, using representations that were pretrained for 30,000 steps (300 seeds; 5-point moving average; shaded regions denote 95% confidence intervals).