Toward Learning POMDPs Beyond Full-Rank Actions and State Observability

Seiji Shaw[1], Travis Manderson[1], Chad Kessens[2], and Nicholas Roy[1]

[1] MIT Computer Science and Artificial Intelligence Lab,
51 Vassar St, Cambridge, MA 02139, USA
{seijis, travislm}@mit.edu, nickroy@csail.mit.edu
[2] DEVCOM Army Research Laboratory
2800 Powder Mill Road, Adelphi, MD 20783, USA
chad.c.kessens.civ@army.mil

Abstract. We are interested in enabling autonomous agents to learn and reason about systems with hidden states, such as locking mechanisms. We cast this problem as learning the parameters of a discrete Partially Observable Markov Decision Process (POMDP). The agent begins with knowledge of the POMDP's actions and observation spaces, but not its state space, transitions, or observation models. These properties must be constructed from a sequence of actions and observations. Spectral approaches to learning models of partially observable domains, such as Predictive State Representations (PSRs), learn representations of state that are sufficient to predict future outcomes. PSR models, however, do not have explicit transition and observation system models that can be used with different reward functions to solve different planning problems. Under a mild set of rankness assumptions on the products of transition and observation matrices, we show how PSRs learn POMDP matrices up to a similarity transform, and this transform may be estimated via tensor decomposition methods. Our method learns observation matrices and transition matrices up to a partition of states, where the states in a single partition have the same observation distributions corresponding to actions whose transition matrices are full-rank. Our experiments suggest that explicit observation and transition likelihoods can be leveraged to generate new plans for different goals and reward functions after the model has been learned. We also show that learning a POMDP beyond a partition of states is impossible from sequential data by constructing two POMDPs that agree on all observation distributions but differ in their transition dynamics.

Keywords: POMDPs · Continual Learning · Task Planning

1 Introduction

When planning and acting in the real world, intelligent agents must learn and reason about hidden information. Of great inspiration to us is the work of Baum et al. [11], which shows that a real autonomous robot can infer a cabinet's locking mechanism from a hypothesis set of mechanisms through interaction. We are

interested in a symbolic variant of the problem where autonomous agents must learn, through interaction, the dynamics of a system with hidden states, without any knowledge of the system state and transitions beforehand. The agent should also compute explicit estimates of transition and observation likelihoods to support downstream operations that manipulate the model, such as task specification to direct agent behavior. Our problem is modeled as learning the parameters of a discrete Partially Observable Markov Decision Process (POMDP) from a sequence of actions and observations acquired through random exploration.

One common approach to learning a representation of a probabilistic latent-variable model like a POMDP is to apply a spectral decomposition to a matrix that encodes correlations of the observable random variables [23, 10]. For POMDPs, spectral methods may be applied to a *Hankel matrix*, which represents the correlation between past and future observations conditioned on a sequence of past and future actions. The decomposition of this matrix can be used to derive a (linear) Predictive State Representation [12, 10]. The 'state' of the PSR is a sufficient statistic that can be used to predict the likelihood of future observations given a possible sequence of actions. This prediction capability allows PSRs to be used as black-box models for reinforcement learning [33, 53]; however, transition and observation likelihoods cannot be directly recovered from a PSR. This lack of interpretability makes these models difficult to manipulate, like changing the goal or reward function for planning. If a goal state of the agent changes, then the PSR must be relearned for the new task, since the underlying state cannot be accessed.

There are other POMDP-learning algorithms that yield estimates of the full model, e.g. observation and transition likelihoods, but under assumptions that ultimately restrict the class of POMDPs that can be learned. Approaches introduced by Azizzadenesheli et al. [7] and Guo et al. [21] utilize tensor decompositions to recover observation distributions for each action whose transition matrix is full-rank. To recover the transitions, however, these approaches must also make the assumption that for each action, the corresponding diagonal observation matrices must be unique for every state, which implies every state has a unique observation distribution. While, full-rank transitions are common when modeling many real-world POMDPs, especially when actions may 'fail' with some probability, many real-world systems have *aliased states*, e.g. states that do not have distinct observation distributions associated with every action. Systems that fall in this class include the locking mechanisms of Baum et al. [11] or many standard POMDPs in the literature, like Tiger [29].

We investigate the relationship between PSRs and tensor decomposition methods to learn a broader class of POMDPs than existing tensor methods. A result established by Carlyle and Paz [15] states that PSRs learn transitions and diagonal observation matrices up to an unknown basis. We then reformulate tensor decomposition methods to estimate the unknown basis to recover the original basis. Our modification of tensor decomposition methods for hidden state inference allows us to simultaneously leverage all observation distributions from *all* actions with full-rank transition methods all at once, rather than a per-

action basis like previous approaches [7, 21]. Should the collection of observation distributions of all full-rank actions be unique for each state, like Tiger, we may recover the full POMDP. Should there exist states that share the same set of observation distributions when aggregated across actions, we learn transitions between partitions of states, where states in a single partition share the same observation distributions over all actions. We also show that when restricted to sequential data, learning transition and observation up to observability partitions *cannot be improved*. We construct an example of two POMDPs whose dynamics differ between aliased states but yield the same distribution over all future observations under an arbitrary sequence of actions.

Learning explicit transition and observation matrices is valuable because these models enable reasoning over environment dynamics. Whereas black-box PSRs only provide predictive likelihoods of observation sequences, access to explicit transition matrices and diagonal observation matrices allows for the specification of rewards after the model has been learned. Our experimental results suggest that our method can correctly learn partition-level transitions and observations and that these likelihoods are necessary to correctly direct agent behavior in POMDPs with very noisy observations.

2 Problem Setting

We assume that the ground truth system can be described as a discrete POMDP, a tuple $(\mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{O}, \mathcal{Z}, b_0, R, \gamma)$. The set $\mathcal{S} = \{s^1, s^2, \dots\}$ is a discrete set of states, $\mathcal{A}$ is a discrete set of actions, and $\mathcal{O} = \{o^1, o^2, \dots\}$ is a discrete set of observations. $\mathcal{T} = \{T^a : a \in \mathcal{A}\}$ denotes a set of row-stochastic state transition matrices. The element $T^a_{ij} = \mathrm{P}(s_{t+1} = s^j | s_t = s^i, a_t = a)$ denotes the probability of transition to state s^j from state s^i after taking action a at time t. The set $\mathcal{Z} = \{O^{ao} : (a, o) \in \mathcal{A} \times \mathcal{O}\}$ describes a collection of diagonal matrices, where $O^{ao}_{ii} = \mathrm{P}(o_t = o | s_t = s^i, a_t = a)$ denotes the emission probability of o under action a when *leaving* state s^i. The distribution $b_0 \in \Delta(\mathcal{S})$ describes the distribution over the initial state. The constant $\gamma \in (0, 1)$ is the reward discount factor.

The agent begins acting in a POMDP with access to the action and observation spaces $\mathcal{A}$ and $\mathcal{O}$. Under a uniform, memoryless random exploration policy $a_t \sim \mathrm{Unif}(\mathcal{A})$ for all $t \geq 1$, the agent collects a dataset $\mathcal{D}$, which is a long string of actions and observations $\mathcal{D} = (a_1, o_1, a_2, o_2, \dots)$. From this data, we wish the agent to estimate the number of hidden states $|\mathcal{S}|$, transition matrices $\hat{\mathcal{T}} = \{\hat{T}^a : a \in \mathcal{A}\}$, and diagonal observation matrices $\hat{\mathcal{Z}} = \{\hat{O}^{ao} : (a, o) \in \mathcal{A} \times \mathcal{O}\}$. We may also require the agent to learn a tabular reward R function by including rewards as observations [26]. We evaluate the approach by measuring the error of the estimated model parameters against those of the ground-truth POMDP. Another evaluation gauges the performance of the agent under a planning algorithm after the POMDP is inferred from $\mathcal{D}$. The last is by evaluating the behavior of a planner at user-designated task after model learning.

For notational convenience, we will often need to refer to strings of actions and observations. We call a string $(a_1, o_1, \dots, a_t, o_t)$ that is observed by the agent in the past a *history*, often abbreviated as *hist*. A string $(a_{t+1}, o_{t+1}, \dots, a_n, o_n)$

the agent may observe in the future will be called a *test* [32, 47]. To abbreviate the expression of the likelihood of observations conditioned on the actions, we write

$$\begin{aligned} p(hist, test) &= \mathrm{P}(o_1, \ldots, o_t, o_{t+1}, \ldots, o_n | a_1, \ldots, a_t, a_{t+1}, \ldots a_n), \\ p(hist, s_t = s^i) &= \mathrm{P}(o_1, \ldots, o_t, s_t = s^i | a_1, \ldots, a_t), \\ p(test | s_t = s^i) &= p(o_{t+1}, \ldots, o_n | a_{t+1}, \ldots, a_n, s_{t+1} = s^i) \ \ldots \text{and so on.} \end{aligned}$$

Occasionally, it will be convenient to stack the diagonals of matrices in $\mathcal{Z}$ associated with the same action a into $|\mathcal{S}| \times |\mathcal{O}|$ *row-stochastic matrices* Obs^a, where $\mathrm{Obs}^a_{i,j} = \mathrm{P}(o_t = o^j | s_t = s^i, a_t = a)$. To distinguish between the two, we refer to the collection $\mathcal{Z}$ as *diagonal observation matrices* and the latter matrices $\{\mathrm{Obs}^a : a \in \mathcal{A}\}$ as *row-stochastic observation* matrices.

3 Learning Predictive State Representations

3.1 Forward, Backward, and Hankel Matrices

To estimate systems of hidden state, a natural place to start is to form an array that expresses the correlation between the observable random variables. A *Hankel matrix* is an instance of these arrays that encodes the joint likelihoods of past and future action-observation trajectories. In this section, we derive the Hankel matrix given knowledge of the ground truth POMDP. Our construction starts with two intermediate factors, called the *forward* and *backward* matrix, which we will multiply together to form the Hankel matrix.[3]

The forward matrix **Forw** is an infinite-by-S matrix that expresses the joint likelihood of observing a history and the current state. After choosing a suitable ordering to map histories to indices (histories of length one, then histories of length two, and so on), we may write $\mathbf{Forw}_{hist,i} = p(hist, s_{t+1} = s^i)$ for a given history $hist = (a_1, o_1, \ldots, a_t, o_t)$. In terms of the POMDP matrices $\mathcal{T}$ and $\mathcal{Z}$, a row of the forward matrix can be expressed by right-multiplying the corresponding matrices in the order of the history:

$$\mathbf{Forw}_{hist,:} = b_0 \cdot O^{a_1,o_1} T^{a_1} \cdots O^{a_t,o_t} T^{a_t}. \tag{1}$$

The backward matrix **Back** is an S-by-infinite matrix that represents the *conditional* likelihood of obtaining the observations of a test after executing its actions, e.g. $\mathbf{Back}_{i,test} = p(test | s_{t+1} = s^i)$. If the $test = (a_{t+1}, o_{t+1}, \ldots, a_n, o_n)$, then a column of **Back** can also be computed similarly to the rows of **Forw**:

$$\mathbf{Back}_{:,test} = O^{a_{t+1},o_{t+1}} T^{a_{t+1}} \cdots O^{a_n,o_n} T^{a_n} \cdot \mathbf{1}, \tag{2}$$

[3] This factorized construction is inspired by the construction of a related matrix, called the *System Dynamics Matrix*, by Singh et al. [47]. The entries of the System Dynamics Matrix (SDM) are the likelihood of a given test *conditioned* on a history. Each row of the Hankel matrix is the same as the SDM except scaled by a constant, the likelihood of the history that indexes the row [8].

where **1** is a vector of one in all entries.

The product of the forward and backward matrices results in the Hankel matrix, which we denote as $\mathcal{H}$. The matrix multiplication unconditions and then marginalizes out the intermediate hidden state. Given a history $hist = (a_1, o_1, \ldots, a_t, o_t)$ and a test $test = (a_{t+1}, o_{t+1}, \ldots, a_n, o_n)$, a Hankel matrix entry is the corresponding joint likelihood of receiving a full string of observations conditioned on taking a full string of actions, e.g.

$$\mathcal{H}_{hist,test} = \mathrm{P}(o_1, \ldots, o_n | a_1, \ldots a_n). \tag{3}$$

The Hankel matrix does not refer to the underlying hidden state of the POMDP and can be estimated from action-observation trajectories. If we had a long string of actions and observations $\mathcal{D}_n = (a_1, o_1, \ldots, a_n, o_n)$, the matrix $\mathcal{H}$ could be estimated by the *suffix-history approach*, taking frequency counts of subsequences of increasing lengths [52, 12]:

$$\hat{\mathcal{H}}_{hist,test} = \frac{\sum_{i=1}^{n-L} \mathbb{I}_{(a_i,o_i,\ldots,a_{i+L},o_{i+L})=hist \oplus test}}{\sum_{i=1}^{n-L} \mathbb{I}_{(a_i,\ldots,a_{i+L},)=\mathrm{acts}(hist \oplus test)}} \tag{4}$$

where $\mathrm{acts}(hist \oplus test)$ is the action sequence associated with $hist \oplus test$, $\oplus$ is the concatenation operator, and $L = |hist \oplus test| < n$.

It is important to note that expressing the Hankel matrix as a factorization of **Forw** and **Back** represents the system under a memoryless policy where future actions are independent of previous observations. To correctly estimate the matrix via Eq. (4), the data must also be collected under a memoryless policy. Using a uniformly random exploration policy admits this condition, but a larger class of non-memoryless exploration policies can be used for importance sampling [13].

3.2 Assumptions

Before we discuss our algorithm, we must state some key assumptions. First, we assume that under a memoryless random exploration-policy $a \sim \pi_{\exp}(\mathcal{A}), \pi \in \Delta(\mathcal{A})$[4] (which, in this paper, we take to be uniform), the induced Markov chain $(s_t, a_t, o_t)_{t \geq 0}$ is ergodic. This property causes the visitation distribution over states to converge to a stationary distribution b_π, with nonzero support over the state space, as the agent explores. Ergodic Markov chains are *irreducible*, i.e., every state is reachable with nonzero probability, and *aperiodic*, i.e., cannot be trapped in periodic cycles. We believe these conditions are reasonable. If an agent becomes trapped in some connected component of the system, then the dynamics of that component are the ones that are learned. Robots also often have passive observation that do not change the environment state. The transition dynamics of these actions are loops with period one, which breaks periodicity.

Second, we also assume that **Forw**, when limited to indices corresponding to one fewer than the maximum sequence length, has the same rank as the

[4] $\Delta(\mathcal{A})$ denotes the set of distributions over the discrete set $\mathcal{A}$.

number of states (e.g. is full-rank), and that **Back** is also full-rank. This assumption is strictly weaker than prior work that assumes that all transition matrices T^a and row-stochastic observation matrices Obs^a are full rank for all actions [7, 21]. The implication can be proven by showing that the products $\text{diag}(b_\pi)T^a\text{Obs}^a\,\text{diag}(b_\pi)^{-1}$ and $T^a\text{Obs}^a$ are submatrices of **Forw** and **Back**, respectively. If b_π is a stationary distribution, as implied by our ergodicity assumption, then every entry is positive, so **Forw** and **Back** are full-rank. Many of the standard POMDPs in the literature, which are also included in our experiments (Sec. 5), are counterexamples for the converse of the implication.

Our assumptions have a few consequences on the estimated Hankel matrix. The starting state distribution b_0 at the start of the learning problem, so b_0 has little influence over the Hankel matrix. Instead, the Hankel matrix will take on the stationary distribution b_π as the initial distribution instead. Furthermore, the rank of the resulting Hankel matrix will be *equivalent* to the number of states of the POMDPs in the restricted class that adhere to our assumptions, as opposed to the lower bound as is the case for general POMDPs [47, 24]. A proof of these claims is in Appendix A.1 of the supplemental material.

3.3 Constructing a Linear Predictive State Representation

Suppose we have a Hankel matrix $\mathcal{H}$, estimated in the limit of infinite data. Since an implication of Lemma 2 is that the rank of Hankel matrix $\mathcal{H}$ is equivalent to the number of states of the POMDP, a natural first step of our method (and learning a PSR) is to compute a rank factorization of $\mathcal{H}$ [12, 10]. One way to achieve this factorization is to compute a singular-value decomposition of the Hankel matrix $\mathcal{H} = U\Sigma V^T$, where singular values under a specified threshold (and their corresponding orthogonal vector components) are dropped. The truncated SVD is converted into a *rank factorization* by computing $A = U\Sigma$ to be the left factor and V^T to be the right factor. Crucially, since $A \cdot V^T$ and $\mathbf{Forw}\cdot\mathbf{Back}$ both form rank factorizations of $\mathcal{H}$ (according to assumptions in Sec 3.2), there must exist some invertible transformation P such that $A = \mathbf{Forw} \cdot P$ and $P^{-1} \cdot \mathbf{Back} = V^T$ (see Appendix A.2 in the supplemental material).

Moving one step earlier in the Hankel construction (Sec. 3.1), we can relate transitions, observations, and initial distributions with the rank factors and the Hankel matrix using Eqs. (1) and (2). Let $hists^{ao}$ denote an ordered set of all history indices that end in action-observation pair ao, and $hists^{-ao}$ denote the same set with the same ordering but without the ending pair ao. From Eqs. (1) and (2), we observe for each $a \in \mathcal{A}, o \in \mathcal{O}$:

$$\mathcal{H}_{hists^{ao},:} = \mathbf{Forw}_{hists^{-ao},:} \cdot O^{ao}T^a \cdot \mathbf{Back} = A_{hists^{-ao},:}(P^{-1}O^{ao}T^aP)V^T \quad (5)$$

$$\mathcal{H}_{\varepsilon,:} = b_0^T \cdot \mathbf{Back} = (b_0^T P) \cdot V^T \quad (6)$$

$$\mathcal{H}_{:,\varepsilon} = \mathbf{Forw} \cdot \mathbf{1} = A \cdot (P\mathbf{1}) \quad (7)$$

After applying the Moore-Penrose inverse of A and V^T to Eqs. (5)–(7), we obtain the observation-transition product, initial belief, and final summation vector up to a similarity transform. The transformed initial belief $m_0 = b_0^T P$ from Eq. (6) is called the *initial vector* and the transformed summation vector $m_\infty = P\mathbf{1}$

from Eq. (7) is called the *final vector*. The product $M^{ao} = P^{-1}O^{ao}T^aP$ from Eq. (5) is called a linear PSR update matrix. Together, this collection of matrices and vectors forms a *linear PSR model* [32, 12]. A PSR can be used to compute the likelihood of observations $o_1, o_2, \ldots, o_n$ under actions $a_1, \ldots, a_n$ by computing the product $\mathrm{P}(o_1, \ldots, o_n | a_1, \ldots, a_n) = m_0^T M^{a_1 o_1} \cdots M^{a_n o_n} m_\infty = b_0 P P^{-1} O^{a_1 o_1} T^{a_1} \cdot O^{a_n o_n} T^{a_n} P^{-1} P \mathbf{1}$, with appropriate normalizations for conditional calculations. With a few more details, the argument sketched above is a proof of a result of Carlyle and Paz [15]. The original result given by the authors was for probabilistic automata, and later versions were proved for HMMs [23, 10, 50, 24].

Proposition 1. *Let $\mathcal{H} = AV^T$ be a rank factorization of a Hankel matrix $\mathcal{H}$ with $\mathrm{rank}(\mathcal{H}) = r$ formed from a POMDP with initial state b_π, transition matrices $\{T^a\}$ and diagonal observation matrices $\{O^{ao}\}$. Suppose $m_0, \{M^{ao}, \forall (a,o) \in \mathcal{A} \times \mathcal{O}\}, m_\infty$ are computed as in Eqs. (5)–(7). Then there exists a nonsingular matrix $P \in \mathbb{R}^{r \times r}$ such that $P^{-1}M^{ao}P = O^{ao}T^a$ for all $a \in \mathcal{A}, o \in \mathcal{O}$, $m_0^T P = b_\pi$, and $P^{-1}m_\infty = \mathbf{1}$.*

4 Computing the Similarity Transform

When a reward is part of the observation, the PSR representation M^{ao} can be used to compute policies that maximize the expected reward. However, if we had access to O^{ao} and T^a, we could compute policies for any reward function. To recover O^{ao} and T^a, we need a way to recover the similarity transform P and apply Prop. 1. The observation and transition matrices can be computed from products $O^{ao}T^a$ by taking the sums of the rows to form the diagonal of O^{ao} and then normalizing to form T^a. In many problem scenarios, P is not the identity matrix, since the rank factors computed via SVD are orthogonal matrices, which is not always the case for **Forw** and **Back**. Our approach can recover P up to a certain partition of states, which we introduce with an example.

A running example. Consider the POMDP illustrated in Fig. 1, which is modified from the Float-Reset domain introduced by Littman and Sutton [32]. Like the original, the `float` action transitions the state up and down a line graph and will always emits an observation of `0`. The `reset` action, also identical to the original, deterministically sets the state to the left end of the graph. This action emits an observation of `1` if the state is already in the leftmost state and `0` otherwise. The observations of the `sense` action are the same as `reset`, except each state of the system does not change. We also augment the system with a reward function; the agent obtains +1 reward for executing any action in the state adjacent to the reset state and zero reward otherwise. This system is challenging to learn due to its nontrivial state aliasing. Aside from the two leftmost states (when treating rewards as observations), all other states in this POMDP have the same observation distributions, regardless of the action.

We wish to capture the difficulties of Sense-Float-Reset to discuss the main output of our algorithm in general terms. For arbitrary POMDPs, we group

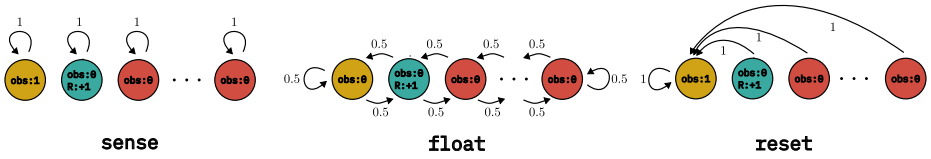

Fig. 1: Sense-Float-Reset. Edges are labeled with transition probabilities, and nodes are labeled with observations and reward received upon *leaving* the state. The reward of a state is zero unless specified otherwise. Node shade denotes observability partitions.

states that have the same observation distribution to form a *partition* of states. We call this grouping an *observability partition*. Of particular importance is the collection of observation distributions that correspond to actions with full-rank transition matrices.

Definition 1 (Full-Rank Observability Partition). *Let $\mathcal{S}_\Pi \subset 2^{\mathcal{S}}$ be a partition of states, such that for any set $S \in S_\Pi$, states $s^i, s^j \in S$ if and only if $\text{Obs}^a_{i,:} = \text{Obs}^a_{j,:}$ for all $a \in \mathcal{A}_{full}$. We call $\mathcal{S}_\Pi$ a full-rank observability partition.*

4.1 Recovery up to a Full-Rank Observability Partition

Our algorithm can estimate the similarity transform P up to the *full-rank observability partition*, which we formalize in Theorem 1. Our statement is given in the regime of infinite data; for parameters introduced for finite data, see Appendix B.1 in the supplemental material.

Theorem 1. *Let $\mathcal{H}$ be a Hankel matrix of POMDP $(\mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{O}, \mathcal{Z}, b_\pi, R, \gamma)$ that adheres to the assumptions in Sec. 3.2, where b_π is the stationary distribution under exploration policy $a_t \sim \text{Unif}(\mathcal{A})$. Let $S_\Pi \subset 2^{\mathcal{S}}$ be the full-rank observability partition of the POMDP. Let m_0, $\{M^{ao} : a \in \mathcal{A}, o \in \mathcal{O}\}$, and m_∞ be the linear PSR model as computed via Eqs. (5)–(7). Then there exists an algorithm on m_0, $\{M^{ao}\}$, and m_∞ that computes a nonsingular matrix $\tilde{P}$, such that if*

$$\tilde{b}_\pi^T = m_0^T \tilde{P} = b_\pi P^{-1} \tilde{P} \tag{8}$$

$$\tilde{O}^{ao} \tilde{T}^a = \tilde{P}^{-1} M^{ao} \tilde{P} = \tilde{P}^{-1} P O^{ao} T^a P^{-1} \tilde{P} \tag{9}$$

$$\tilde{b}_\infty = \tilde{P}^{-1} m_\infty = \tilde{P}^{-1} P \mathbf{1} \tag{10}$$

then

$$\sum_{s^i \in S} \tilde{b_\pi}_i = \sum_{s^i \in S} b_{\pi i} \tag{11}$$

$$\sum_{s^i \in S} (\tilde{b}_\pi^T \tilde{O}^{a_1,o_1} \tilde{T}^{a_1} \dots \tilde{O}^{a_n,o_n} \tilde{T}^{a_n})_i = \sum_{s^i \in S} (b_\pi^T O^{a_1 o_1} T^{a_1} \dots O^{a_n o_n} T^{a_n})_i \tag{12}$$

$$\tilde{b}_\infty = \mathbf{1} \tag{13}$$

for all $a_1, \dots, a_n \in \mathcal{A}$, $o_1, \dots, o_n \in \mathcal{O}$, integer $n > 0$ and partition set $S \in S_\Pi$.

Fig. 2: An illustration of Theorem 1 applied to Sense-Float-Reset. While the individual values inside a single box (state) may not be read as the belief likelihood of the system occupying that state, summing indices over partitions, represented by box shades, will compute the likelihood of the system state in that partition.

What Theorem 1 states is that we must sum over indices of the initial 'belief vector' to compute the likelihood the system is in a particular partition (Eq. (8)). The same remains true when computing joint likelihoods between observations and the current state partition (Eq. (12); see Fig. 2 for a worked example for Sense-Float-Reset). For POMDPs that have unique observation distributions across all actions, each state is in its own singleton partition, and we can recover the full similarity transform. Otherwise, we recover P up to the full-rank observability partition. We note it is possible for us to recover some POMDPs that have fewer observations than states, since the collection of *distributions* over emitted observations across all actions must be distinct (see Appendix C.5 in the supplemental material for examples).

To benefit from the result of Theorem 1, the systems to be learned must satisfy the assumptions discussed in Sec. 3.2 and contain full-rank actions. Full-rank actions commonly arise in many automated manipulation contexts, our domain of interest. In automated manipulation, robot actions have a desired transition state but may also *fail* (a gripper misses a grasp, slips of a drawer handle, etc.). One way these actions have been modeled in robot planning systems is to designate a successful 'desired state' with some success likelihood p_{succ}, and have the system state 'fail' with some likelihood (causing a self-transition) [30, 20]. In POMDP terms, these types of actions can be simply modeled as the convex combination $p_{\text{succ}}T + (1 - p_{\text{succ}})I$, where T is a matrix with rows containing all zeros except for a single entry of 1 (the 'desired states'), the identity I indicates self-loop failure dynamics, and p_{succ} the likelihood of an action succeeding. Under mild assumptions (e.g. $p_{\text{succ}} \neq 1/2, 1$), these actions are full-rank (see Appendix A.6 in supplemental material).

4.2 Recovering Observation Distributions from Full-Rank Actions

We now introduce an algorithm that computes the similarity transform $\tilde{P}$. Our approach is a reformulation of the tensor decomposition method [4, 7] for linear PSR models. Our procedure begins by marginalizing out the observations in matrices M^{ao}, yielding the transitions T^a up to similarity transform P. This marginalization can be done by summing all matrices M^{ao} over all $o \in \mathcal{O}$ for some fixed $a \in \mathcal{A}$:

$$\sum_{o \in \mathcal{O}} M^{ao} = P\left(\sum_{o \in \mathcal{O}} O^{ao} T^a\right) P^{-1} = P T^a P^{-1} \tag{14}$$

With a slight abuse of notation, we denote $P^{-1}T^aP$ as the matrix M^a. The next step of our procedure continues with transitions that are full-rank, which can easily be determined by a threshold test on the singular value decomposition on all matrices M^a. Let $\mathcal{M}_{full} = \{P^{-1}T^aP : a \in \mathcal{A}_{full}\}$ be the set of full-rank transitions. Next, we compute the diagonal observation matrices associated with the full-rank actions. For each $M^a \in M_{full}$ and $o \in \mathcal{O}$ we compute

$$M^{ao} \cdot M^{a^{-1}} = PO^{ao}T^aP^{-1}(PT^aP^{-1})^{-1} = PO^{ao}P^{-1}. \tag{15}$$

Since we know that all matrices O^{ao} are diagonal, the eigenvalues of the matrices $M^{ao}M^{a^{-1}}$ will be the diagonal entries of O^{ao}. If the entries of a particular O^{ao} are unique, then the eigenvectors computed from an eigendecomposition of $M^{ao}M^{a^{-1}}$ will recover the columns of P up to a scalar factor. However, it is common to have repeated observation likelihoods across states for a single action (like all of Sense-Float-Reset), and an eigendecomposition may produce *any* spanning set of the invariant space corresponding to the repeated eigenvalue.

To reduce ambiguity, we wish to compute a *joint diagonalization* of all matrices $M^{ao}M^{a^{-1}}$, which attempts to diagonalize each matrix with the same similarity transform. We apply a method of He et al. [22]. Their method exploits the fact that *sums* of matrices $\{M^{ao}M^{a^{-1}} : a \in \mathcal{A}_{full}, o \in \mathcal{O}\}$ do not change the invariant spaces spanned by eigenvectors of each matrix $M^{ao}M^{a^{-1}}$. Suppose $\{w^{ao} : a \in \mathcal{A}_{full}, o \in \mathcal{O}\}$ is a set of weights, then the weighted sum

$$\sum_{a\in\mathcal{A}_{full},o\in\mathcal{O}} w^{ao}M^{ao}M^{a^{-1}} = P\Big(\sum_{a\in\mathcal{A}_{full},o\in\mathcal{O}} w^{ao}O^{ao}\Big)P^{-1} \tag{16}$$

is still diagonalizable by P. Should we choose *random* weights w_{ao}, then the eigenvalues will be distinct up to states that share the same observation distribution almost surely. We sample these weights from the unit sphere $\mathbb{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|-1}$ [22].

Lemma 1. *Let weights* $\{w_{ao} : a \in \mathcal{A}_{full}, o \in \mathcal{O}\}$ *be sampled i.i.d. with respect to* $\mathrm{Unif}(\mathbb{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|-1})$ *and* $\Lambda = \sum_{a\in\mathcal{A}_{full},o\in\mathcal{O}} w^{ao}O^{ao}$. *Then* $\Lambda_{ii} = \Lambda_{jj}$ *with prob. 1 if and only if* $O^{ao}_{ii} = O^{ao}_{jj}$ *for all* $o \in \mathcal{O}$ *and all* $a \in \mathcal{A}_{full}$.

When multiple states have the same observation distribution for all actions, the eigenvalues corresponding to those states will be the same, so their eigenvectors cannot be uniquely determined. Thus, the similarity transform P' is nonunique when we have a nontrivial full-rank observability partition, the consequences of which we discuss in the next session.

4.3 Recovering Partition-Level Belief Likelihoods and Transitions

The recovered similarity transform P' formed by the eigenvectors of the random sum in Eq. (16), but not the partition-level transitions. When the full-rank observability partition is nontrivial, the matrix $Q = P^{-1}P'$ is block-diagonal, with invertible blocks that correspond to states within the same partition (see Appendix A.4 in supplemental material for a proof). This matrix Q prevents us from using P' as the similarity transform promised in Theorem 1. For example,

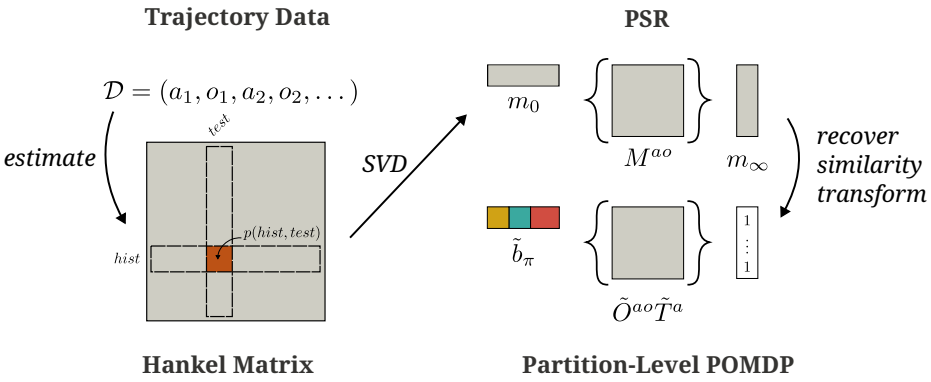

Fig. 3: An illustration of the entire algorithm pipeline described from Sec. 3.1-4.

when applying P' as a similarity transform to the PSR vector m_0, a restriction to the subindices of the partition S_1 yields $[m_0 P']_{S_1} = [b_0^T P^{-1} P']_{S_1} = [b_0^T]_{S_1} Q_1$, so the sum of the entries is not a proper likelihood, violating Eq. (11) of Theorem 1.

To recover partition-level likelihoods and transitions, we look to the final vector of the linear PSR after applying the transform P', e.g. $Pm_0 = P'^{-1}P\mathbf{1}$. Intuitively, by applying $\text{diag}(Pm_0)$ as a similarity transform, we transform the final vector back to $\mathbf{1}$, recapturing a marginalization of the latent state variable. To avoid scenarios where $P'^{-1}m_0$ has entries of zero, we perform a pre-processing step by multiplying the system with a random block-diagonal rotation matrix R, whose blocks correspond to the full-rank observability partition. We take the transform $\text{diag}(RP'^{-1}m_\infty)RP'^{-1}$ as the similarity transform $\tilde{P}$ that satisfies Theorem 1 (see Appendix A.5 for the proof of correctness and runtime).

Summary A diagram that sketches the entire learning algorithm can be found in Fig. 3. Prop. 1 and Theorem 1 suggest that we can compute a PSR and then recover the corresponding system model by recovering the unknown similarity transform P. Sections 4.2 and 4.3 present an algorithm that applies to all POMDPs that admit the assumptions of Sec. 3.2, and computes P up to the full-rank observability partition of states. The set of POMDPs learned by our method captures a large class of real-world POMDPs that arise in manipulation scenarios.

5 Experiments

Our experiments evaluate the fidelity of the learned POMDP models and explore the advantages of estimating transition and observation likelihoods. We seek to know, empirically, how quickly the learned partition-level transition and observation matrices converge to ground truth values. We also wish to know how the performance of the planning model is impaired by estimation error from little data. Lastly, we evaluate whether the transition and observation likelihood estimates can be leveraged to specify a reward function to elicit desired behavior from a planner. All experiments are compared against linear PSRs and

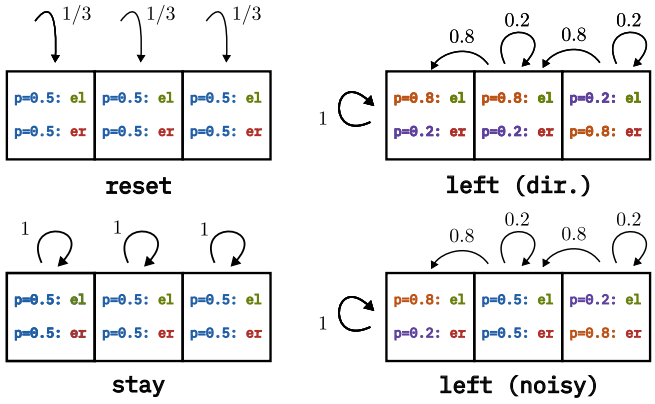

Fig. 4: The directional and noisy hallway domains. The observations `el` and `er` represent 'end-right' and 'end-left' respectively. Transitions are shared and observation distsributions differ in the middle state in the `left` and `right` actions. The `right` action is the mirror image of `left`, with the obs. likelihoods swapped in the middle state in the directional environment.

Expectation-Maximization (EM) [39, 45] with a number of states determined by the number of components of the truncated SVD when learning a linear PSR.

For our planning experiments, we verify our approach on two standard domains, Tiger [29] and T-Maze (with a single corridor state) [9], and Sense-Float-Reset. To allow the agent to collect an arbitrary-length string of data in all domains, we modify T-Maze to choose the next state randomly from the initial state distribution instead of terminating the sequence of interactions. In the supplemental material, Appendices B.1 and C contain details on the parameters of the learning algorithm and planner, including sensitivity analyses to algorithm parameters. Rewards of the original POMDPs have been learned as observations for planning. For our reward-specification experiments, we introduce two novel domains (*noisy hallway* and *directional hallway*) whose observation and transition matrices can be fully recovered by our method (see Fig. 4). For more details on all domains, see Appendix C.5 in the supplemental material.

Convergence to true POMDP parameters. In Fig. 5, our results suggest that our method successfully recovers the underlying observation models through the L_1 error of learned observation and partition-level transition likelihoods against ground truth. EM consistently converges to a local minimum that correctly predicts future observations but obtains incorrect POMDP matrices.

Planning performance with the learned model. To evaluate the performance of the learned models, we apply a standard solver to the original ground truth POMDPs, learned PSRs, and learned POMDPs. We use the sampling-based planning approach PO-UCT of Silver and Veness [46] with the correction described by Shah et al. [43]. Ideally, average planning performance should be the same across ground truth models, PSRs, and the partition-level POMDPs,

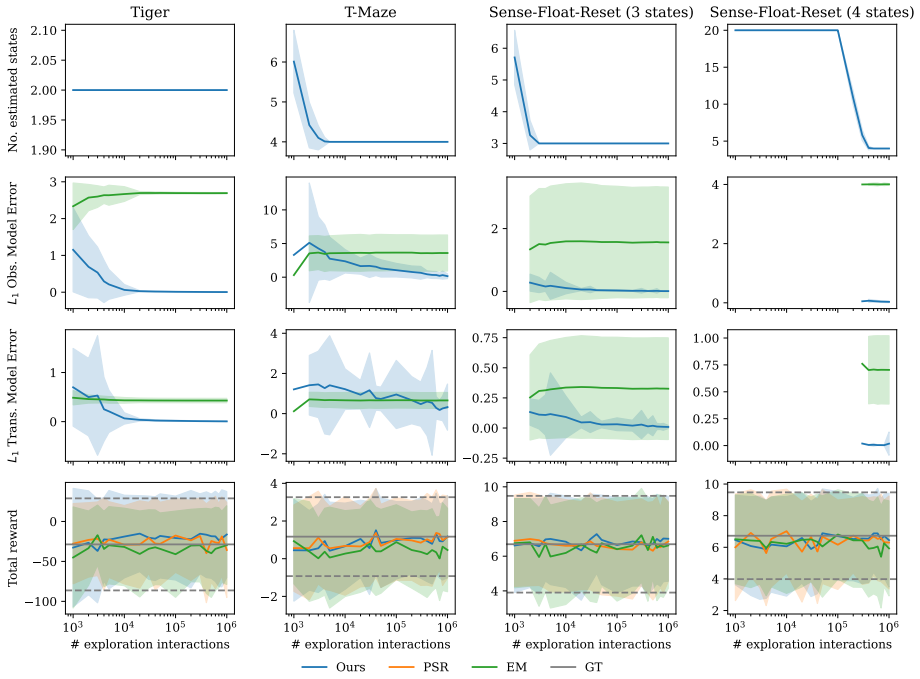

Fig. 5: Error bars represent standard deviation over 100 seeds. The y-axis is scaled to make convergence visible. **Row 1:** Estimated number of states. **Row 2:** Obs. matrix error relative to ground truth. **Row 3:** Trans. matrix error. This error is only measurable once the estimated number of states matches that of ground truth, which truncates the curves. **Row 4:** Total reward from planner under different models.

since they all learn the same distribution over observations and rewards given a potential sequence of actions (see Appendix C.3 for rollout strategies for each model). Figure 5 reports that the performance across models is roughly the same.

Planning performance on specified rewards. We explore whether the likelihoods and observations yielded by our algorithm can be leveraged to direct agent behavior by specifying a reward function. When explicit POMDPs are available, we can analyze the learned observation matrices to directly find the states to assign positive reward. In the past, if a PSR did not learn a reward model, then rewards were determined by observations [12]. Otherwise, the entire model must be relearned to estimate a reward model that depends on state [26].

Our evaluations of this experiment are carried out on the two noisy hallway domains, where we attempt to direct the agent to drive the POMDP to the 'middle' hallway state with ambiguous observations (say, to collect more data for a poorly functioning perception model). We compare the strategies of assigning rewards to observations and assigning rewards to states. In the directional domain, we assign +1 reward to action-observation pairs (`left`, `end-left`) and (`right`, `end-right`) for the former strategy, and assign +1 reward to the state whose maximum likelihood observations under `left` and `right` are their

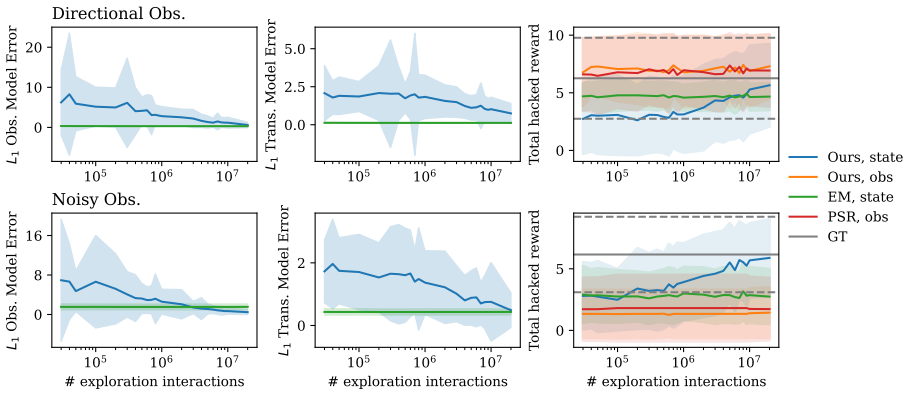

Fig. 6: The agent receives +1 reward for each timestep in the designated goal state (middle of hallway). 'Obs' refers to assigning rewards to action-observation pairs, whereas 'state' refers to assigning rewards to states. Error bars report standard deviation over 100 seeds. 'Hacked reward' refers to planning performance under a new reward fun.

corresponding hallway ends for the latter. For the noisy environment, we reward the same action-observation pairs as the directional environment and also (`left`, `end-right`) and (`right`, `end-left`) for the former strategy, and add +1 reward to the state that maximizes the sum of entropy of observation distributions across all actions for the latter. The former strategy is evaluated on PSRs and POMDPs, whereas the latter is evaluated on learned POMDP models only. Performance is judged on total reward gathered under the new reward function.

Results can be found in Fig. 6. In the directional domain, models that use the first strategy allow the planner to drive the middle state because it is easily identified by the observations received under `left` and `right`. The second strategy performs poorly with less data due to slow convergence of transition matrices (see Appendix C.4). In the noisy domain, the uniform belief state and belief state that places all mass on the middle of the hallway yield the same observation mixture distribution when weighted by the belief over states. The planner that uses observation-based rewards sees that playing the `reset` action or controlling the state to stay in the center to yield the same rewards, leading to unnecessary `reset` actions. The planner that uses the rewards emitted from the highest-entropy state correctly rewards the middle state after the transition matrices begin to converge, eliciting the correct behavior. This additional flexibility highlights that learning POMDPs maintains all the advantages of PSRs and obtains the flexibility to exploit observation and transition likelihood models.

6 On the Necessity of Observability Partitions

While learning POMDPs up to observability partitions falls short of our goal of recovering all possible POMDPs with arbitrary transition and observation matrices, the restriction of the observability partitions is in fact the best we can achieve from a single sequential trajectory. We will show that there are multiple POMDPs with different dynamics within the observability partitions that

yield the same distribution over observations regardless of the actions taken, even when we assume that **Forw** and **Back** are full-rank. The principal issue is that there are multiple valid 'forward-backward' factorizations of the same Hankel matrix. An equivalent statement is that there is a similarity transform P that converts one valid forward-backward factorization into another. Our counterexample to identifiability is framed in terms of the last point: we provide a POMDP and a similarity transformation that, when applied in the manner of Prop. 1 P, yields a valid b'_0, O'^{ao} and T'^{a} of a different POMDP.

The counterexample is a perturbation of the *sense-float-reset* with three states (Fig. 1). We perturb the `float` transition matrix, which we associate with a new action `float'`, to allow for the states to jump between the two ends of the graph. The `reset`, `sense`, and observation matrices remain unchanged.

$$T^{\texttt{float}'} = \begin{pmatrix} .4 & .5 & .1 \\ .5 & .1 & .4 \\ 0 & .5 & .5 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & .95 & .05 \end{pmatrix} \tag{17}$$

It can be readily verified that after applying the matrix P as a similarity transform on all POMDP matrices of the perturbed sense-float-reset problem, that we obtain another valid POMDP. Under a random exploration policy, the Hankel matrices computed from data trajectories of both POMDPs are the same, in the limit of infinite data (see Appendix A.7 in the supplemental material for proof).

Theorem 2. *Let $\mathcal{M} = (b_0, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{Z})$ and $\mathcal{M}' = (b'_0, \mathcal{A}', \mathcal{T}', \mathcal{O}', \mathcal{Z}')$ be the two POMDPs corresponding to the POMDPs as presented above, with arbitrary initial distributions b_0 and b'_0. Suppose the Hankel matrix $\mathcal{H} = \lim_{n\to\infty} \hat{\mathcal{H}}$, where $\hat{\mathcal{H}}$ is computed via Eq.* (4) *under POMDP $\mathcal{M}$, and suppose $\mathcal{H}'$ is the analogous Hankel matrix under $\mathcal{M}'$. Then $\mathcal{H}$ and $\mathcal{H}'$ are equivalent.*

We can conclude that the distribution over observations is the same, regardless of the action sequences taken in either POMDP, since these observation distributions can be read from the row of either Hankel matrix indexed by the empty history. What our counterexample suggests is that knowledge of the internal structure of a POMDP is *not necessary* to predict future outcomes of the system. While being able to predict future trajectories may be sufficient for planning (with goals set as a function of observations), the ambiguity of internal structure prevents us from fine-grain model manipulation such as setting goals on specific latent states for planning. Learning exact models that can be leveraged in this way can only be done when certain conditions are met.

7 Related Work

Much of the theory surrounding Hankel matrix methods has been developed to study Hidden Markov Models (HMMs), which can be viewed as POMDPs with a single action. Hankel factorization approaches to learn HMMs have been rediscovered many times [25, 10, 27]. Since then, conditions under which the Hankel matrix represents a finite-state HMM have been investigated [5, 50]. Sample complexity and runtime complexity of these methods have also been studied in

many contexts, including from sample trajectories of the HMM [24, 44] and from directly querying Hankel entries [34].

The study of POMDP-learning has been informed by the development of Hankel methods. PSRs [32, 47, 12, 52] were discovered separately, but share their mathematical basis with Hankel factorization approaches to Hidden Markov Models above. In the learning theory community, these spectral approaches have largely been focused on learning predictive models for model-based reinforcement learning [28, 17, 33, 16], but not learning explicit transition and observation distributions. The study of *exact* HMM and POMDP recovery has largely been done using tensor decomposition methods, which require full-rankness assumptions on transition and row-stochastic observation matrices [37, 24, 7, 21]. Relaxing these assumptions is the focus of this paper.

Other approaches to learning POMDPs (and HMMs) have also been explored. Unlike the stochastic case, POMDPs have been shown to be learnable when actions and observations are deterministic by matching states as equivalence classes of observed histories that produce the same outputs on future actions [6, 40, 18, 14, 41]. The Expectation-Maximation algorithm has been applied to both POMDPs and HMMs [39, 45], but are often stuck in local minima. Mixed-integer programs to search through automata [49] and inductive logic schemes [3] have also been applied. An alternate view of learning deterministic discrete systems has been studied from the lens of combinatorial filters [42]. Recurrent deep-learning-based architectures have also been explored [51, 2, 1].

8 Conclusion and Future Work

We present a method that learns the parameters of a discrete POMDP from an action-observation sequence gathered under a random exploration policy up to a partition of the state space. Our approach applies tensor decomposition methods to estimate a similarity that to recover a full POMDP model under a milder set of assumptions relative to prior work. In domains where each state has a unique observation distribution aggregated across all full-rank actions, we recover the true POMDP. Otherwise, we learn the transitions between full-rank observability partitions of the state space. We also show that we can only distinguish POMDPs up to observability partitions from one sequential data trajectory.

One limitation of our work is the assumption that the forward and backward matrices are full-rank. In general, it is known that the rank of the Hankel matrix can be strictly smaller than the minimal number of states of an HMM, which is also a POMDP [27, 50]. While we have learned much from Hankel factorization algorithms for POMDP learning, the presence of this rank discrepancy suggests a broader algorithmic framework may be required to better understand the problem. Another limitation is that the algorithm only learns models from domains with few states due to the size of the requisite Hankel matrix. Prior work on HMMs characterizes computational efficiency based on the minimum size of the Hankel matrix required to ensure full-rankness [24, 44]. Generalizing these results to POMDPs may help us scale our algorithms to larger systems.

Acknowledgments. We thank Prof. Leslie Pack Kaelbling and Prof. Tomás Lozano-Pérez for their constructive feedback on early versions of the formulation of the problem studied in this paper. Our use of large language models is solely limited to the completion of routine coding tasks, such as experiment launching and plotting code. This work was supported by ARL under Grant W911NF-23-2-0012, ONR under Grant N00014-22-1-267, the NSF Graduate Research Fellowship Program under Grant 2141064, and the Siegel Family Quest for Intelligence. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DEVCOM Army Research Laboratory, the Office of Naval Research, the National Science Foundation, the U.S. Government, or the Siegel Family Quest for Intelligence.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agarwal, A., Alomar, A., Alumootil, V., Shah, D., Shen, D., Xu, Z., Yang, C.: PerSim: Data-Efficient Offline Reinforcement Learning with Heterogeneous Agents via Personalized Simulators. In: Advances in Neural Information Processing Systems. vol. 34, pp. 18564–18576 (2021)
2. Allen, C., Kirtland, A., Tao, R.Y., Lobel, S., Scott, D., Petrocelli, N., Gottesman, O., Parr, R., Littman, M., Konidaris, G.: Mitigating Partial Observability in Sequential Decision Processes via the Lambda Discrepancy. In: Advances in Neural Information Processing Systems. vol. 37, pp. 62988–63028 (2024)
3. Amir, E., Chang, A.: Learning Partially Observable Deterministic Action Models. Journal of Artificial Intelligence Research **33**, 349–402 (2008)
4. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., Telgarsky, M.: Tensor decompositions for learning latent variable models. Journal of Machine Learning Research **15**(80), 2773–2832 (2014)
5. Anderson, B.D.O.: The Realization Problem for Hidden Markov Models. Mathematics of Control, Signals and Systems **12**(1), 80–120 (1999)
6. Angluin, D.: Learning regular sets from queries and counterexamples. Information and Computation **75**(2), 87–106 (1987)
7. Azizzadenesheli, K., Lazaric, A., Anandkumar, A.: Reinforcement Learning of POMDPs using Spectral Methods. In: Proceedings of the Conference on Learning Theory. pp. 193–256 (2016)
8. Bacon, P.L., Balle, B., Precup, D.: Learning and Planning with Timing Information in Markov Decision Processes. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (2015)
9. Bakker, B.: Reinforcement Learning with Long Short-Term Memory. In: Advances in Neural Information Processing Systems. vol. 14 (2001)
10. Balle, B., Carreras, X., Luque, F.M., Quattoni, A.: Spectral learning of weighted automata. Machine Learning **96**(1), 33–63 (2014)
11. Baum, M., Bernstein, M., Martin-Martin, R., Höfer, S., Kulick, J., Toussaint, M., Kacelnik, A., Brock, O.: Opening a lockbox through physical exploration. In: Proceedings of the IEEE-RAS International Conference on Humanoid Robotics. pp. 461–467 (2017)

12. Boots, B., Siddiqi, S.M., Gordon, G.J.: Closing the learning-planning loop with predictive state representations. The International Journal of Robotics Research **30**(7), 954–966 (2011)
13. Bowling, M., McCracken, P., James, M., Neufeld, J., Wilkinson, D.: Learning predictive state representations using non-blind policies. In: Proceedings of the International Conference on Machine Learning. pp. 129–136 (2006)
14. Brafman, R.I., De Giacomo, G.: Regular Decision Processes: A Model for Non-Markovian Domains. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 5516–5522 (2019)
15. Carlyle, J., Paz, A.: Realizations by stochastic finite automata. Journal of Computer and System Sciences **5**(1), 26–40 (1971)
16. Chen, F., Bai, Y., Mei, S.: Partially Observable RL with B-Stability: Unified Structural Condition and Sharp Sample-Efficient Algorithms. In: Proceedings of the International Conference on Learning Representations (2022)
17. Chen, F., Wang, H., Xiong, C., Mei, S., Bai, Y.: Lower Bounds for Learning in Revealing POMDPs. In: Proceedings of the International Conference on Machine Learning. pp. 5104–5161 (2023)
18. Dean, T., Angluin, D., Basye, K., Engelson, S., Kaelbling, L., Kokkevis, E., Maron, O.: Inferring finite automata with stochastic output functions and an application to map learning. Machine Learning **18**(1), 81–108 (1995)
19. Ding, J., Rhee, N.H.: When a Matrix and Its Inverse Are Nonnegative. Missouri Journal of Mathematical Sciences **26**(1), 98–103 (2014)
20. Garrett, C.R., Paxton, C., Lozano-Pérez, T., Kaelbling, L.P., Fox, D.: Online Replanning in Belief Space for Partially Observable Task and Motion Problems. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 5678–5684 (2020)
21. Guo, Z.D., Doroudi, S., Brunskill, E.: A PAC RL Algorithm for Episodic POMDPs. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. pp. 510–518 (2016)
22. He, H., Kressner, D., Plestenjak, B.: Randomized methods for computing joint eigenvalues, with applications to multiparameter eigenvalue problems and root finding. Numerical Algorithms (2024)
23. Hsu, D., Kakade, S.M., Zhang, T.: A spectral algorithm for learning Hidden Markov Models. Journal of Computer and System Sciences **78**(5), 1460–1480 (2012)
24. Huang, Q., Ge, R., Kakade, S., Dahleh, M.: Minimal Realization Problems for Hidden Markov Models. IEEE Transactions on Signal Processing **64**(7), 1896–1904 (2016)
25. Ito, H., Amari, S.I., Kobayashi, K.: Identifiability of hidden Markov information sources and their minimum degrees of freedom. IEEE Transactions on Information Theory **38**(2), 324–333 (1992)
26. Izadi, M.T., Precup, D.: Point-Based Planning for Predictive State Representations. In: Advances in Artificial Intelligence. pp. 126–137. Springer, Berlin, Heidelberg (2008)
27. Jaeger, H.E.: Discrete-time, Discrete-valued Observable Operator Models: A Tutorial. Technical Report, European Research Consortium for Informatics and Mathematics (1998)
28. Jin, C., Kakade, S., Krishnamurthy, A., Liu, Q.: Sample-Efficient Reinforcement Learning of Undercomplete POMDPs. In: Advances in Neural Information Processing Systems. vol. 33, pp. 18530–18539 (2020)
29. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101**(1), 99–134 (1998)

30. Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning in belief space. The International Journal of Robotics Research **32**(9–10), 1194–1227 (2013)
31. Lee, J.M.: Sard's theorem. In: Introduction to Smooth Manifolds, pp. 125–149. Springer (2012)
32. Littman, M., Sutton, R.S.: Predictive Representations of State. In: Advances in Neural Information Processing Systems. vol. 14 (2001)
33. Liu, Q., Chung, A., Szepesvari, C., Jin, C.: When Is Partially Observable Reinforcement Learning Not Scary? In: Proceedings of the Conference on Learning Theory. pp. 5175–5220 (2022)
34. Mahajan, G., Kakade, S., Krishnamurthy, A., Zhang, C.: Learning Hidden Markov Models Using Conditional Samples. In: Proceedings of the Conference on Learning Theory. pp. 2014–2066 (2023)
35. Meckes, E.S.: The Random Matrix Theory of the Classical Compact Groups. Cambridge Tracts in Mathematics, Cambridge University Press (2019)
36. Moitra, A.: Algorithmic Aspects of Machine Learning. Cambridge University Press (2018)
37. Mossel, E., Roch, S.: Learning nonsingular phylogenies and hidden Markov models. The Annals of Applied Probability **16**(2), 583–614 (2006)
38. Norris, J.R.: Discrete-Time Markov Chains. Cambridge University Press (1997)
39. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
40. Rivest, R.L., Schapire, R.E.: Inference of Finite Automata Using Homing Sequences. Information and Computation **103**(2), 299–347 (1993)
41. Ronca, A., Licks, G.P., Giacomo, G.D.: Markov Abstractions for PAC Reinforcement Learning in Non-Markov Decision Processes. In: Proceedings of the International Joint Conference on Artificial Intelligence. vol. 4, pp. 3408–3415 (2022)
42. Sakcak, B., Timperi, K.G., Weinstein, V., LaValle, S.M.: A mathematical characterization of minimally sufficient robot brains. The International Journal of Robotics Research **43**(9), 1342–1362 (2024)
43. Shah, D., Xie, Q., Xu, Z.: Nonasymptotic Analysis of Monte Carlo Tree Search. Operations Research **70**(6), 3234–3260 (2022)
44. Sharan, V., Kakade, S.M., Liang, P.S., Valiant, G.: Learning Overcomplete HMMs. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
45. Shatkay, H., Kaelbling, L.P.: Learning Geometrically-Constrained Hidden Markov Models for Robot Navigation: Bridging the Topological-Geometrical Gap. Journal of Artificial Intelligence Research **16**, 167–207 (2002)
46. Silver, D., Veness, J.: Monte-Carlo Planning in Large POMDPs. In: Advances in Neural Information Processing Systems. vol. 23 (2010)
47. Singh, S., James, M., Rudary, M.: Predictive State Representations: A New Theory for Modeling Dynamical Systems. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence. pp. 512–519 (2004)
48. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cumberland, UNITED STATES (2005)
49. Toro Icarte, R., Klassen, T.Q., Valenzano, R., Castro, M.P., Waldie, E., McIlraith, S.A.: Learning reward machines: A study in partially observable reinforcement learning. Artificial Intelligence **323**, 103989 (2023)
50. Vidyasagar, M.: The complete realization problem for hidden Markov models: A survey and some new results. Mathematics of Control, Signals, and Systems **23**(1), 1–65 (2011)

51. Wang, A., Li, A.C., Klassen, T.Q., Icarte, R.T., Mcilraith, S.A.: Learning Belief Representations for Partially Observable Deep RL. In: Proceedings of the International Conference on Machine Learning. pp. 35970–35988 (2023)
52. Wolfe, B., James, M.R., Singh, S.: Learning predictive state representations in dynamical systems without reset. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 980–987 (2005)
53. Zhan, W., Uehara, M., Sun, W., Lee, J.D.: PAC Reinforcement Learning for Predictive State Representations. In: Proceedings of the International Conference on Learning Representations (2022)

A Appendix: Omitted Proofs

A.1 Formalization of Assumptions Consequences

Lemma 2. *Let $(\mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{O}, \mathcal{Z}, b_0, R, \gamma)$ be a POMDP. Suppose that the agent has collected a trajectory $\mathcal{D}_n = (a_1, o_1, \ldots, a_n, o_n)$ for $n > 0$, where $a_i \sim \mathrm{Unif}(\mathcal{A})$ for all i. Suppose the POMDP admits the assumptions outlined above. Furthermore, let $\hat{\mathcal{H}}$ be the 'empirical' Hankel matrix as computed in Eq. (4). Consider the Hankel matrix in the 'limit of infinite data,' where $\mathcal{H} = \lim_{n\to\infty} \hat{\mathcal{H}}$. Then $\mathcal{H}$ is the Hankel matrix of POMDP $(\mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{O}, \mathcal{Z}, b_\pi, R, \gamma)$ and $\mathrm{rank}(\mathcal{H}) = |\mathcal{S}|$.*

Here, we prove the consequences of the assumtions discussed in Section 3.2. For our proof, we rely on a fundamental result on the convergence of ergodic Markov chains to stationary distributions.

Theorem 3 (Ergodic Theorem, [38]). *Let T be ergodic with stationary distribution π, and let b_0 be any initial distribution. Let $(X_n)_{n\geq 0}$ be a Markov chain with respect to T with initial distribution b_0. Then, for any bounded function $f : \mathcal{S} \to \mathbb{R}$ we have*

$$\mathrm{P}\left(\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n-1} f(X_k) = \bar{f}\right) = 1$$

where

$$\bar{f} = \sum_{i\in\mathcal{S}} \pi_i f(i),$$

and b_π is the stationary distribution of T.

We also require knowledge of the stationary distribution of Markov chains created as a 'sliding window' of another ergodic Markov chain.

Lemma 3. *Let $(Y_t)_{t\geq 0}$ be a Markov chain with ergodic transition matrix T, with stationary distribution π, over state space $\mathcal{Y} = \{1, \ldots, k\}$. Then the Markov chain $(Y_t, Y_{t+1}, \ldots Y_{t+n-1})_{t\geq 0}$ is also ergodic, with stationary distribution $\tilde{\pi}(i_1, \ldots i_n) = \pi_{i_0} T_{i_0,i_1} \ldots T_{i_{n-2},i_{n-1}}$.*

Proof. The transitions of this Markov chain can be expressed as

$$\mathrm{P}(i_{t+n}, \ldots, i_{t+1} | i_{t+(n-1)}, \ldots, i_t) = T_{i_{t+n-1}, i_{t+n}}$$

and zero if the indices do not follow the 'sliding window' form above. We verify the stationary distribution claimed in the conclusion of the statement. Suppose that $(j_1, \ldots, j_n)$ is a state of the Markov chain $(Y_t, Y_{t+1}, Y_{t+n-1})$. When we apply the transition likelihoods above, we find that

$$\begin{aligned}
&\sum_{i_1,\ldots,i_n\in\mathcal{Y}} \tilde{\pi}(i_1, \ldots i_n)\, \mathrm{P}(Y_{t+1} = j_1, \ldots, Y_{t+n} = j_n | Y_t = i_1, \ldots, Y_{t+n-1} = i_n) \\
&= \sum_{i_1} \pi(i_1) T_{i_1,j_1} \ldots, T_{j_{n-1},j_n} \\
&= \pi(j_1) T_{j_1,j_2} \ldots T_{j_{n-1},j_n} = \tilde{\pi}(j_1, \ldots, j_n).
\end{aligned}$$

The second line applies the definition of a transition above and the proposed stationary distribution, and the third line uses the fact that π is the stationary distribution of T.

We now have all the tools we need to prove Lemma 2.

Proof. **The Hankel matrix takes on the stationary distribution b_π as the stationary distribution**:

We first consider the stochastic process $X_t = (s_t, a_t, o_t)$, which is a Markov chain due to the factorization structure of a POMDP [48]. Per our assumptions in Section 3.2, we assume that the Markov chain $(X_t)_{t\geq 0}$ over the state space $\mathcal{X} = \{(s^i, o^j, a^k) \in \mathcal{S} \times \mathcal{O} \times \mathcal{A} : \mathrm{P}(o^k|s^i a^k) > 0\}$ is ergodic. Suppose that its stationary distribution is p_π. We denote $p_\pi(s, a, o)$ to be the likelihood of the Markov chain (s, a, o) under p_π. Of interest is the marginal stationary distribution of the *POMDP* state $s \in \mathcal{S}$, under p_π, which we denote as the vector b_π, where $(b_\pi)_i = \sum_{a\in\mathcal{A}, o\in\mathcal{O}} p_\pi(s, a, o)$.

We observe that for any $(a_0, o_0, s_0, \ldots, a_{k-1}, o_{k-1}, s_{k-1}) \in \mathcal{X}^k$, under Lemma 3 and Theorem 3,

$$\begin{aligned}&\lim_{n\to\infty} \frac{1}{n-k} \sum_{i=k}^{n} \mathbb{I}_{a_0,o_0,s_0,\ldots,a_{k-1},o_{k-1},s_{k-1}=X_{i-k},\ldots,X_i} \\ &= p_\pi(a_0, o_0, s_0) \prod_{i=0}^{k-1} \mathrm{P}(X_{i+1} = s_{i+1}, o_{i+1}, a_{i+1}|X_i = s_i, o_i, a_i)\end{aligned}, \tag{18}$$

almost surely for any integer $k \geq 1$. Let P corresponds to the law induced by the stationary distribution p_π and transition and observation models of the POMDP. We will now use Eq. (18) as a way to understand the convergent values of the Hankel matrix.

Let $\bar{\mathcal{D}}_n = (s_1, a_1, o_1, \ldots, s_n, a_n, o_n)$ be a sequence of the induced Markov chain, and let $\mathcal{D}_n = (a_1, o_1, \ldots, a_n, o_n)$ be the same dataset with the state variable omitted. Let $hist = (a^{j_1}, o^{k_1}, \ldots, a^{j_t}, o^{k_t})$ and $test = (a^{j_{t+1}}, o^{k_{t+1}}, \ldots, a^{j_L}, o^{k_L})$ be action-observation sequences, length $L < n$. Then, we may evaluate the empirical Hankel matrix $\hat{\mathcal{H}}$ using Eq. (18).

$$\begin{aligned}&\hat{\mathcal{H}}_{hist,test} \\ &= \frac{\sum_{i=1}^{n-L} \mathbb{I}_{(a_i,o_i,\ldots,a_{i+L},o_{i+L})=hist\oplus test}}{\sum_{i=1}^{n-L} \mathbb{I}_{(a_i,\ldots,a_{i+L},)=(a^{j_1},\ldots,a^{j_L})}} \\ &= \frac{1/(n-L)}{1/(n-L)} \\ &\quad \cdot \frac{\sum_{s^{m_1},\ldots,s^{m_L}} \sum_{i=1}^{n-L} \mathbb{I}_{(a_i,o_i,s_i\ldots,a_{i+L},o_{i+L},s_{i+L})=(a^{j_1},o^{k_1},s^{m_1},\ldots,a^{j_L},o^{k_L},s^{m_L})}}{\sum_{o^{k_1}s^{k_1},\ldots,o^{k_L}s^{k_L}} \sum_{i=1}^{n-L} \mathbb{I}_{(a_i,o_i,s_i,\ldots,a_{i+L},o_{i+L},s_{i+L})=(a^{j_1},o^{k_1},s^{m_1},\ldots,a^{j_L},o^{k_L},s^{m_L})}}\end{aligned}$$

Taking the limit of n to infinity on both sides and applying Eq. (18) (noting the denominator is nonzero almost surely under a uniform random exploration

policy) yields

$$\begin{aligned}
&\lim_{n\to\infty} \hat{\mathcal{H}}_{hist,test} \\
&= \mathrm{P}(o^{k_1}, \ldots, o^{k_L} | a^{j_1}, \ldots a^{j_L}) \\
&= \sum_{s^{m_1},\ldots,s^{m_L},s^{m_{L+1}}} \mathrm{P}(s^{m_1}, o^{k_1}, \ldots, s^{m_L}, o^{k_L}, s^{m_{L+1}} | a^{j_1}, \ldots a^{j_L}) \\
&= \sum_{s^{m_1},\ldots,s^{m_L},s^{m_{L+1}}} \mathrm{P}(s^{m_1}) \, \mathrm{P}(o^{k_1}, s^{m_2} | s^{m_1} a^{m_1}) \cdots \mathrm{P}(s^{m_{L+1}}, o^{k_L} | s^{m_L} a^{j_L}) \\
&= b_\pi O^{a^{j_1} o^{k_1}} T^{a^{j_1}} \cdots O^{a^{j_L} o^{k_L}} T^{a^{j_L}} \cdot \mathbf{1}
\end{aligned}$$

The second line unmarginalizes the state, and the third line factorizes the full likelihood in terms of the POMDP's transition and observation conditional likelihoods. The last line unpacks the probability law P introduced by Eq. (18) back into matrix notation. We can see that splitting the product for the last line above over *hist* and *test* will reproduce individual rows and columns of **Forw** and **Back**, respectively. Finally, we observe that **Forw** has taken on the distribution b_π as the initial vector in the product.

The Hankel matrix is full-rank: Since we know a submatrix of **Forw** formed a subselection of rows is full-rank then the full forward matrix **Forw** is full-rank as well. Thus, we know that both of the **Back** and **Forw** are full-rank. This means we can find $|S|$ linearly-independent rows of **Back** and $|S|$ linearly-independent columns of **Forw**, which we assemble into submatrices K and W respectively. We observe, then, that K and W are full-rank and square. The product $K \cdot W$, then, must also be full-rank and square. When we multiply $\mathbf{Forw} \cdot \mathbf{Back} = \mathcal{H}$, we observe, then, that $K \cdot W$ is a submatrix of $\mathcal{H}$. Then we know that

$$|S| = \mathrm{rank}(K \cdot W) \leq \mathrm{rank}(\mathcal{H}) \leq \min(\mathrm{rank}(\mathbf{Forw}), \mathrm{rank}(\mathbf{Back})) = |S|,$$

so $\mathrm{rank}(H) = |S|$.

A.2 Proof of Proposition 1

We have two rank factorizations of $\mathcal{H}$: $\mathbf{Forw} \cdot \mathbf{Back} = A \cdot V^T = \mathcal{H}$. Since all matrix factors involved are full-rank, we may take the Moore-Penrose inverse of **Forw** and **Back**, which results in $\mathbf{Forw}^\dagger A \cdot V^T \mathbf{Back}^\dagger = I$. Then $\mathbf{Forw}^\dagger A$ is nonsingular and $V^T \mathbf{Back}^\dagger$ is its inverse.

We take the product $(\mathbf{Forw}^\dagger A)$ to be P. A consequence of the assumptions in Section 3.2 is that $A_{hists^{-ao},:}$ is full-rank for all $a \in \mathcal{A}$ and $o \in \mathcal{O}$. Thus, we could have repeated the argument above, but replacing A with $A_{hists^{-ao}}$ and $\mathcal{H}$ with $\mathcal{H}_{hists^{-ao},:}$, and find that $P = \mathbf{Forw}^\dagger_{hists^{-ao},:} A_{hists^{-ao}}$.

What remains is to show that we can apply P to recover the POMDP initial belief, diagonal observation matrices, transition matrices, and final summing vector from the linear PSR models. Let $a \in \mathcal{A}$ and $o \in \mathcal{O}$. Following Eq. (5),

we know that $M^{ao} = A^{\dagger}_{hists^{-ao},:} \cdot \mathcal{H}_{hists^{ao},:} \cdot {V^T}^{\dagger}$. If we apply P as a similarity transform, we find that

$$\begin{aligned} P^{-1}M^{ao}P &= P^{-1}A^{\dagger}_{hists^{-ao},:} \cdot \mathcal{H}_{hists^{ao},:} \cdot {V^T}^{\dagger}P \\ &= P^{-1}A^{\dagger}_{hists^{-ao},:} \cdot \mathbf{Forw}_{hists^{-ao},:} \cdot O^{ao}T^a \cdot \mathbf{Back} \cdot {V^T}^{\dagger}P \\ &= P^{-1}P \cdot O^{ao}T^aP^{-1}P = O^{ao}T^a \end{aligned}$$

The initial belief vector b_0 and all ones vector $\mathbf{1}$ can be recovered from the initial vectors m_0 and m_∞ in the same manner (they only feature an inversion of P on either the left or right sides, but not both as above). □

A.3 Proof of Lemma 1

The 'only if' direction is immediate. We prove the 'if' direction by proving its contrapositive.

Fix i, j such that $1 \leq i < j \leq |\mathcal{S}|$. Suppose that there exists an $a \in \mathcal{A}_{full}$, $o \in \mathcal{O}$ such that $O^{ao}_{ii} \neq O^{ao}_{jj}$. Let $(ao)_1, \ldots, (ao)_{|\mathcal{A}_{full}|\cdot|\mathcal{O}|}$ be an ordering on $\mathcal{A}_{full} \times \mathcal{O}$. Let a, b be two $|\mathcal{A}_{full}| \cdot |\mathcal{O}|$-dimensional vectors such that $a_k = O^{(ao)_k}_{ii}$ and $b_k = O^{(ao)_k}_{jj}$ for all $1 \leq k \leq |\mathcal{A}_{full}| \cdot |\mathcal{O}|$. Then we know that $a \neq b$.

Consider the event that $w \in \mathbb{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|}$, such that $\Lambda_{ii} = \sum_k^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|} w_k O^{(ao)_k}_i i$ and $\Lambda_{jj} = \sum_k^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|} w_k O^{(ao)_k}_j j$ are equivalent. Written in terms of the notation introduced above, we have that $\langle a - b, w \rangle = 0$. This means that w must be contained in the hyperplane $H = \{x \in \mathbb{R}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|} : \langle x, a - b \rangle = 0\}$, which also passes through the origin. We recognize that $H \cap \mathbb{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|}$ is a $|\mathcal{A}_{full}| \cdot |\mathcal{O}| - 2$-dimensional submanifold (a lower-dimensional sphere) of $\mathcal{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|}$. We know that the measure of this submanifold under the induced uniform measure on $\mathbb{S}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|-1}$ from the Lebesgue measure on $\mathbb{R}^{|\mathcal{A}_{full}|\cdot|\mathcal{O}|-1}$ is zero [31]. Thus, the probability of sampling w so that $\Lambda_{ii} = \Lambda_{jj}$ is zero as well. Therefore, the complement of this event, that $\Lambda_{ii} \neq \Lambda_{jj}$, must occur with probability one. □

Remark 1. By a similar argument above, we obtain that $\mathrm{diag}(\Lambda) \neq \mathbf{0}$ with probability 1, where $\mathbf{0}$ is the zero vector. The argument replaces the discussion of the vector $a - b$, as constructed above, with the individual vectors a or b.

A.4 Proof that Similarity Transform is Recovered up to Block-Diagonal Matrix

First, we formalize the claim made in Sec. 4.3.

Lemma 4. *Let P' be the similarity transform as determined by an eigendecomposition of the random sum of Eq.* (16)*. Without loss of generality, permute the columns of P' and P so that states in the same full-rank observability partition are in consecutive indices. Then*

$$P^{-1}P' = \begin{pmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & & Q_k \end{pmatrix} \tag{19}$$

where the blocks $Q_i \in \mathbb{R}^{|S_i| \times |S_i|}$ *are nonsingular w.p. 1, where* $|S_i|$ *is the* i^{th} *partition in the permuted index ordering.*

Let X denote the random sum as expressed in Eq. (16), and let $P\Lambda P^{-1}$, $P'\Lambda P'^{-1}$ be the two diagonalizations as discussed in Section 4.2. Let $S_\Pi = \{S_1, \ldots, S_k\}$ be the full-rank observability partition. By Lemma 1, then $\Lambda_{ii} = \Lambda_{jj}$ for all s^i, s^j in the same partition as $S \in S_\Pi$. Furthermore, by the remark in Section A.3, we know that $\text{diag}(\Lambda) \neq \mathbf{0}$ with probability 1, where $\mathbf{0}$ is the zero vector.

Suppose the indices of these matrices are ordered as stated in the hypothesis. Since $X = P\Lambda P^{-1} = P'\Lambda P'^{-1}$, then we have that $P'^{-1}P\Lambda = \Lambda P'^{-1}P$ (e.g. $P'^{-1}P$ and Λ commute). Examining the entries of equation $P'^{-1}P\Lambda - \Lambda P'^{-1}P = 0$ yields that $(\Lambda_{ii} - \Lambda_{jj})(P'^{-1}P)_{ij} = 0$. If s^i and s^j are contained in separate partitions, then $\Lambda_{ii} - \Lambda_{jj} \neq 0$, so $P'^{-1}P_{ij} = 0$. Thus, $P'P^{-1}$ has the desired block-diagonal structure. Since we know that both P^{-1} and P' are invertible, so must be $P'^{-1}P$. Thus, we know the blocks are invertible as well. □

A.5 Proof of Theorem 1

Section 4.3 claims that applying the matrix $P'R\,\text{diag}(R^TP'^{-1}m_\infty)$ is a similarity transformation $\tilde{P}$ that satisfies the implication of Theorem 1. As a reminder, P' are the eigenvectors from the eigendecomposition of the matrix in Eq. (16) and R is a random block-diagonal rotation matrix with the same block structure as $PP'^{-1} = Q$ (Lemma 4), whose blocks are distributed over the Haar measure over the corresponding copy of $SO(n)$.

We begin our argument by first applying the similarity transformation $\tilde{P}$ to a learned PSR m_0, $\{M^{ao} : a \in \mathcal{A}, o \in \mathcal{O}\}$ and m_∞. We find that

$$m_0\tilde{P} = b_\pi QR\,\text{diag}(R^TQ^{-1}\mathbf{1}) \tag{20}$$

$$\tilde{P}^{-1}M^{ao}\tilde{P} = \text{diag}(R^TQ^{-1}\mathbf{1})^{-1}R^TQ^{-1} \cdot (T^aO^{ao}) \cdot QR\,\text{diag}(R^TQ^{-1}\mathbf{1}) \tag{21}$$

$$\tilde{P}^{-1}m_\infty = \text{diag}(R^TQ^{-1}\mathbf{1})^{-1}R^TQ^{-1} \cdot \mathbf{1} \tag{22}$$

If we unpack the block structure of Q, R, and $\text{diag}(R^TQ^{-1}m_\infty)$ in Eqs. (20) and (22), we find

$$[m_0\tilde{P}]_{S_i} = b_\pi Q_iR_i\,\text{diag}(R_i^TQ_i^{-1}[\mathbf{1}]_{S_i}) \tag{23}$$

$$[\tilde{P}^{-1}m_\infty]_{S_i} = \text{diag}(R_i^TQ_i^{-1}[1]_{S_i})^{-1}R_i^TQ_i^{-1} \cdot [\mathbf{1}]_{S_i} \tag{24}$$

where R_i and Q_i are the blocks associated with the full-rank observability partition S_i, and $[U]_\mathcal{I}$ represents the values of the vector or matrix-valued quantity U indexed by a set of indices $\mathcal{I}$.

First, we must justify that the relations expressed in Eqs. (20)–(24) are well-defined. We already know R and Q are nonsingular. We must then show all entries of the vector $R^TQ^{-1}\mathbf{1}$ are nonzero to allow for the existence of $\text{diag}(R^TQ^{-1}\mathbf{1})$. This fact is a consequence of known properties of the Haar measure over special orthogonal matrices [35, Section 1.2]. Fix a full-rank observability partition S_i. It is known that corresponding rotation matrix blocks R_i^T and R_i are identically distributed with respect to the Haar measure on $SO(|S_i|)$ [35, pg. 18]. Furthermore, since Q_i is nonsingular, we know that $Q_i^{-1}\mathbf{1}$ is not the zero vector. Thus, it is also known that the random vector $R_i^T \cdot (Q_i^{-1}[\mathbf{1}]_{S_i})$ is uniformly distributed over the $(|S_i|-1)$-sphere with radius $\left\|Q_i^{-1}[\mathbf{1}]_{S_i}\right\|_2$ [35, pg. 19-20, 26]. By the same argument discussed in the proof of Lemma 1, the entries of $R_i^TQ_i^{-1}[\mathbf{1}]_{S_i}$ must be nonzero with probability one. By taking a union bound over all full-rank observability partitions, *all* entries of $R^TQ^{-1}\mathbf{1}$ must be nonzero with probability one as well.

What remains is to prove the correctness of the relations Eqs. (11)–(13) in Theorem 1. The expression that we obtain $\tilde{P}^{-1}m_\infty = \mathbf{1}$ is immediate from Eqs. (22) and (24). First, we justify Eq. (11). Fix an full-rank observability partition S_i. Then

$$\begin{aligned}\sum_{i\in S_i}[\tilde{b}_\pi]_i &= [m_0\tilde{P}]_{S_i}^T \cdot [\mathbf{1}]_{S_i} \\ &= [b_\pi^T]_{S_i}Q_iR_i\,\text{diag}(R_i^TQ_i^{-1}[\mathbf{1}]_{S_i}) \cdot \text{diag}(R_i^TQ_i^{-1}[\mathbf{1}]_{S_i})^{-1}R_i^TQ_i^{-1} \cdot [\mathbf{1}]_{S_i} \\ &= [b_\pi^T]_{S_i} \cdot [\mathbf{1}]_{S_i} \\ &= \sum_{i\in S_i}[b_\pi]_i\end{aligned}$$

The second line applies Eqs. (23) and (24). The proof for Eq. (12) is nearly the same, and can be reached by deriving an analagous expression to Eq. (23) by first multiplying out the corresponding sequence of matrices $\tilde{P}^{-1}M^{ao}P$ and unpacking the block structure of Q and R again. □

A.6 Proof of Full-Rank Transition Claim

We formally state the claim made in the deliberation of Section 4.1

Proposition 2. *Let T be an $n \times n$ matrix, with rows that are all zeros except for a single entry of 1 per row. Let $p \in [0,1)$, and $p \neq 1/2$. Then the convex combination $pT + (1-p)I$ is nonsingular.*

Proof. We first observe that the proof is immediate if $p = 0$, so we focus on the case for $p \in (0,1)$. Suppose, for the sake of contradition, that there exists $v \in \mathbb{R}^n$ such that matrix-vector product $(pT + (1-p)I)\,v = 0$. This must be true if and only if

$$Tv = \left(\frac{p-1}{p}\right)v,$$

or that v is an eigevector of T with eigenvalue $(p-1)/p$.

We claim that the eigenvalues of T are either zero or roots of unity. If this claim is true, we arrive at a contradiction, because if $p \neq 1/2$ and $p \in (0,1)$, then $(p-1)/p$ cannot be equal -1.

We prove this claim by induction on the number of rows and columns. As the base-case, we take a 1×1 "matrix" $\begin{pmatrix}1\end{pmatrix}$. The eigenvalue of this matrix is unity. Next, we assume that the claim holds for $m \times m$ matrices with rows of all zeros except for a single one. Suppose we have a $(m+1) \times (m+1)$ matrix T' of the same structure. If T' is a permutation matrix, then we know its eigenvalues are roots of unity [19], so suppose that T' is not a permutation matrix. Then T' must have at least one columns that is all zeros. We then examine the characteristic polynomial $\phi(T')$. Without loss of generality, assume that column is the first column of the matrix. Then, we can write out the expression for the characteristic polynomial $\phi(T')$:

$$\begin{aligned}
\phi(T') &= \det\left(T' - \lambda I\right) \\
&= \det\left(\begin{array}{c|c} -\lambda & \cdots \\ \hline 0 & T'_{2:,2:} - \lambda I \end{array}\right) \\
&= -\lambda \cdot \det(T'_{2:,2:} - \lambda I) \\
&= -\lambda \cdot \phi(T'_{2:,2:})
\end{aligned}$$

where $T'_{2:,2:}$ is the $m \times m$ submatrix of T' that omits the first row and column of T'. This submatrix is also a matrix with all zeros for every row except for a single one, since we eliminated a column of only zeros from T'. Thus, we know that the eigenvalues of T' are 0 and the eigenvalues of $T'_{2:,2:}$, which, by the induction hypothesis, are also zero and roots of unity. □

A.7 Proof of Theorem 2

We first begin by characterizing the stochastic process $\{s_t\}_{t=0}^{\infty}$, the marginals of states variables during exploration ($a_t \sim \text{Unif}(\mathcal{A})$ i.i.d.). We begin by proving the following lemma:

Lemma 5. *The stochastic process $\{s_t\}_{t\geq 0}$ is a Markov chain, and its associated transition matrix is*

$$T = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} T^a.$$

Proof. We first begin with conditional joint likelihood that we would like to factorize and unmarginalize present and past actions. Because actions are sampled

i.i.d., we have that:

$$
\begin{aligned}
&\mathrm{P}(s_t, s_{t+2}|s_{t+1}) \\
&= \sum_{a_t, a_{t+1} \in \mathcal{A}} \mathrm{P}(s_t, s_{t+2}|a_t, s_{t+1}, a_{t+1}) \, \mathrm{P}(a_t|s_{t+1}) \, \mathrm{P}(a_{t+1}|s_{t+1}) \\
&= \sum_{a_t, a_{t+1} \in \mathcal{A}} \mathrm{P}(s_t|a_t, s_{t+1}) \, \mathrm{P}(s_{t+2}|s_{t+1}, a_{t+1}) \, \mathrm{P}(a_t|s_{t+1}) \, \mathrm{P}(a_{t+1}|s_{t+1}) \\
&= \sum_{a_t, a_{t+1} \in \mathcal{A}} \mathrm{P}(s_t, a_t|s_{t+1}) \, \mathrm{P}(s_{t+2}, a_{t+1}|s_{t+1}) \\
&= \mathrm{P}(s_t|s_{t+1}) \, \mathrm{P}(s_{t+2}|s_{t+1})
\end{aligned}
$$

The third line applies the fact that $s_{t+2} \perp s_t | s_{t+1}, a_t, a_{t+1}$ and $s_i \perp a_{i+1} | s_{i+1}$ due to the conditional independencies induced by a POMDP.

We then observe that the transition likelihood between two states is the averaged transition likelihood across all actions.

$$
\mathrm{P}(s_{t+1}|s_t) = \sum_{a_t \in \mathcal{A}} \mathrm{P}(s_{t+1}|s_t, a_t) \, \mathrm{P}(a_t|s_t) = \sum_{a_t \in \mathcal{A}} \frac{\mathrm{P}(s_{t+1}|s_t)}{|\mathcal{A}|} \tag{25}
$$

The last equality invokes that $a_t \sim$ Unif i.i.d. Thus, we can conclude from Eq. (25) that the transition matrix of this Markov chain T can be written as the average of the transition matrices of the POMDP, e.g. $T = \sum_{a \in \mathcal{A}} T^a / |\mathcal{A}|$. □

We also observe that the stationary distribution of this Markov chain is b_π, as defined in the proof of Lemma 2, Appendix A.1 and is unique (otherwise, we would violate ergodicity of the Markov chain $\{s_t, a_t, o_t\}$).

Now, we are ready to prove the main claim of Theorem 2. Let the transition and diagonal observation matrices of the original POMDP be $\mathcal{T} = \{T^a : a \in \mathcal{A}\}$ and $\mathcal{Z} = \{O^{ao} : a \in \mathcal{A}, o \in \mathcal{O}\}$, and suppose that $\mathcal{T}' : \{T^{a\prime} : a \in \mathcal{A}\}$ and $\mathcal{Z}' : \{O^{ao\prime} : a \in \mathcal{A}, o \in \mathcal{O}\}$ are the transition and diagonal matrices of the POMDP after transformation P is applied. Since b_π is the unique stationary distribution of Markov chain $\{s_t\}_{t \geq 0}$ evolving according to POMDP with transition matrices $\mathcal{T}$, then b_π must be a unique right-eigenvector of matrix $T = \sum_{a \in \mathcal{A}} T^a / |\mathcal{A}|$ with eigenvalue 1 [38]. Therefore, the row-stochastic vector $b'_\pi = b_\pi P^{-1}$ must be a right vector of the transition matrix $T' = \sum_{a \in mcA} T'^a / |\mathcal{A}|$ because

$$
\begin{aligned}
&b_\pi'^T \left(\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} T'^a \right) \\
&= b_\pi P^{-1} \left(\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} P T^a P^{-1} \right) \\
&= b_\pi \left(\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} T^a \right) P^{-1} \\
&= b_\pi P^{-1} \\
&= b'_\pi
\end{aligned}
$$

From the relation above, we conclude that b'_π must be a unique stationary distribution for the stochastic process $\{s'_t\}_{t \geq 0}$ that evolves according to POMDP transition matrices $\mathcal{T}'$.

Lastly, fix $hist = (a_1, o_1, \ldots, a_t, o_t)$ and $test = (a_t, o_t, \ldots, a_n, o_n)$. We evaluate Eqs. (1)–(3) to relate Hankel matrices $\mathcal{H}$ and $\mathcal{H}'$:

$$
\begin{aligned}
\mathcal{H}_{hist,test} &= b_\pi T^{a_1} O^{a_1 o_1} T^{a_2} O^{a_2 o_2} \cdots T^{a_n} O^{a_n o_n} \mathbf{1} \\
&= b_\pi P^{-1} P T^{a_1} P^{-1} P O^{a_1 o_1} P^{-1} P T^{a_2} P^{-1} \cdots P T^{a_n} P^{-1} P O^{a_n O_n} P^{-1} P \mathbf{1} \\
&= b'_\pi T^{a_1\prime} O^{a_1 o_1\prime} T^{a_2\prime} O^{a_2 o_2\prime} \cdots T^{a_n\prime} O^{a_n o_n\prime} \mathbf{1} \\
&= \mathcal{H}'_{hist,test}.
\end{aligned}
$$

The third line involes the fact that the provided similarity transform in the perturbed sense-float-reset counterexample (Eq. (17)) has rows that sum to one. Thus, we conclude that $\mathcal{H} = \mathcal{H}'$. □

B Appendix: Additional Algorithmic Details

B.1 Parameters Introduced for Finite Data

Our derivations so far have assumed to be in the asymptotic regime where we have made perfect estimates of the Hankel matrix. In practice, with finite data, we only have the empirical Hankel matrix, $\hat{\mathcal{H}}$, which is subject to random perturbations. Naturally, some adjustments to the calculations expressed in the previous section must be made to account for estimation error. There are three operations where estimation error influences the learning procedure: rank estimation via truncated SVD, determining full rank transition matrices M^a, and obtaining the observability partition to compute the random matrix R in Sec. 4.3. For the Hankel matrix rank, we find that introducing a threshold on the low-rank approximation's reciprocal condition number to be sufficient. To test for transition full-rankness, we found that a minimum singular value $\sigma_{\min}$ threshold was acceptable. To find the full-rank observability partition, we consider two

Algorithm 1 Learn-POMDP

Require: Dataset $\mathcal{D} = (a_1 o_1, a_2 o_2 \dots)$, reciprocal cond. number $1/\kappa$, substring length L, minimum trans. mat singular value σ_{min}, observation sim. threshold τ_{obs}:

1: $substrings \leftarrow \{(a_i o_i a_{i+1} o_{i+1} \dots a_{i+k} o_{i+k})\}_{i=1}^{|\mathcal{D}|/2-L}$
2: $\mathcal{H} \leftarrow \text{EstimateHankel}(substrings)$ ▷ Entries estimated via Eq. 4.
3: $U, \Sigma, V^T \leftarrow \text{TruncatedSVD}(\mathcal{H}, r, 1/\kappa)$
4: $A \leftarrow U\Sigma$
5: $m_0, \{M^{ao} : a \in \mathcal{A}, o \in \mathcal{O}\}, m_\infty \leftarrow \text{ComputePSR}(A, V^T, \mathcal{H})$ ▷ via Eqs. 5-7.
6: $M_{obs} = []$
7: **for** $a \in \mathcal{A}$ **do**
8: $\quad M^a \leftarrow \sum_{o \in \mathcal{O}} M^{ao}$
9: $\quad$ **if** $\text{MinSingularValue}(M^a) > \sigma_{min}$ **then** ▷ Detect full-rank actions.
10: $\quad\quad$ **for** $o \in \mathcal{O}$ **do**
11: $\quad\quad\quad (M_{obs}).\text{append}(M^{ao}(M^a)^{-1})$
12: $w_1, \dots w_{|M_{obs}|} \sim \text{Unif}(\mathbb{S}^{|M_{obs}|-1})$
13: $P' \leftarrow \text{Eigenvectors}(\sum_{i=1}^{|M_{obs}|} w_i (M_{obs})_i)$ ▷ via Eq. 16. Eigenvectors form columns of P'.
14: **for** $M^{ao}(M^a)^{-1} \in M_{obs}$ **do**
15: $\quad O^{ao} \leftarrow P'^{-1} M^{ao} (M^a)^{-1} P'$
16: $[S_1, \dots, S_k] \leftarrow \text{DetectPartitions}(\{O^{ao}\}, \tau_{obs})$ ▷ via procedure in Sec. B.1.
17: **for** $S_i \in [S_1, \dots, S_k]$ **do**
18: $\quad R_i \sim \text{Unif}(SO(|S_i|))$
19: $R \leftarrow \text{BlockDiag}([R_1, \dots, R_k], [S_1, \dots, S_k])$
20: $\tilde{P} \leftarrow P' R \,\text{diag}(R^T P'^{-1} m_\infty)$ ▷ Blocks are specified by indices in partitions $S_1, \dots, S_k$.
21: $\tilde{b} \leftarrow m_0 \tilde{P}$
22: **for** $(a, o) \in \mathcal{A} \times \mathcal{O}$ **do**
23: $\quad \tilde{O}^{ao} \tilde{T}^a \leftarrow \tilde{P} M^{ao} \tilde{P}^{-1}$

Ensure: $\tilde{b}$, $\{\tilde{O}^{ao}\tilde{T}^a : a \in \mathcal{A}, o \in \mathcal{O}\}$

observation distributions to be equivalent if their L^1 norm falls below a threshold τ_{obs}.

While the transition and observation likelihoods computed from the data will converge to the values true values asymptotically, approximation error prevents us from directly reading the parameter estimates as probabilities [21, 7]. Before using the learned model we project all parameters back to the probability simplex by minimizing the L_2 norm by quadratic programming.

Algorithm pseudocode can be found in algorithm 1. We note that for all experiments, while it is theoretically correct to construct a block-diagonal rotation matrix R by the full-rank observability partition as stated above, we find in practice it is sufficient to multiply by fully-dense random rotation matrix R' after computing the initial SVD (line 4). This modification uses AR' and $R'^T V$ to compute the linear PSR and takes $R = I$ instead at line 19. We still require a τ_{obs} parameter to compute the partition-level transition errors in Figs. 5 and 6.

B.2 Runtime Complexity in Floating-Point Operations

The runtime of our approach, which we measure in floating-point operations, is dominated by the rank factorization of the Hankel matrix and computation of the PSR update matrices. We define the *full-observability length* of a POMDP (and notate as n^{obs}) to be the smallest length of histories and tests so that the Hankel matrix, whose rows and columns are indexed by action-observation sequences enumerated up to this length, is full-rank. Suppose we are given a Hankel matrix that enumerates histories up to $n^{obs}+1$ in the rows (so that $A_{hists^{-ao},:}$ in Eq. (5) is full-rank) and tests up to length n^{obs} in the columns. The size of the Hankel matrix, then, must be $O((|\mathcal{A}||\mathcal{O}|)^{n^{obs}+1}) \times O((|\mathcal{A}||\mathcal{O}|)^{n^{obs}+1})$. Computing the truncated SVD with an appropriately set singular value, threshold, then, has runtime $O(|\mathcal{S}| \cdot (|\mathcal{A}||\mathcal{O}|)^{2(n^{obs}+1)})$. To compute the PSR update matrices, we pseudoinvert the right rank factor once, which also has complexity $O(|S| \cdot (|\mathcal{A}||\mathcal{O}|)^{2(n^{obs}+1)})$. Then, to compute each M^{ao}, we must pseudoinvert the right factor $A_{hists^{-ao},:}$, which has runtime $O(|\mathcal{S}| \cdot (|\mathcal{A}||\mathcal{O}|)^{2n^{obs}})$, and then compute the product $A^{\dagger}_{hists^{-ao},:}\mathcal{H}_{hists^{ao},:}(V^T)^{\dagger}$, which has runtime

$$O\left(|\mathcal{S}| \cdot (|\mathcal{A}||\mathcal{O}|)^{2n^{obs}+1}\right) + O\left(|\mathcal{S}|^2(|\mathcal{A}||\mathcal{O}|)^{n^{obs}+1}\right).$$

Putting everything together, we have a full runtime of:

$$O\left(|\mathcal{S}|(|\mathcal{A}||\mathcal{O}|)^{2(n^{obs}+1)} + |\mathcal{S}|^2(|\mathcal{A}||\mathcal{O}|)^{n^{obs}+2}\right) \tag{26}$$

Interestingly, our calculation suggests a runtime that is polynomial when the observability length n^{obs} scales favorably when as the number of states of a particular class of POMDPs increases, which aligns with prior work on Hidden Markov Models [24]. Further investigation of computational tractability learning framework (e.g. PAC-learning) for the multi-action case would be an interesting direction of future work.

C Appendix: Additional Experimental Details

C.1 Algorithm Parameter Selection

As discussed in Appendix B, the behavior performance of our learning algorithm depends on a few manually-specified parameters. This section reviews all the parameters that must be specified to run our approach, and the parameters values selected for our experiments (for a summary, see Table 1).

For Hankel estimation, of practical concern is the selection of the size Hankel matrix to estimate, or the sequences to include as row and column indices. Like many other approaches [23, 10], we use every possible action-observation sequence up to a certain length. We expose this length as an algorithm parameter. While smaller lengths will result in faster convergence of matrix entry estimates,

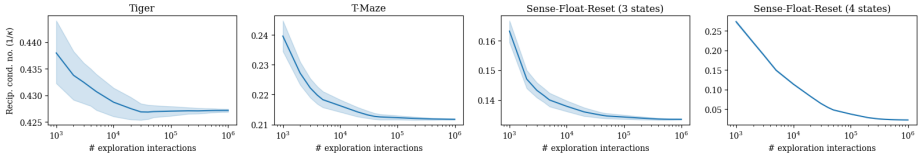

Fig. 7: The ratio of the rth condition number over the largest condition number of Hankel matrices as the amount of observed data increases, where $r = |S|$ is the number of states of the POMDP. Each plot is averaged over 100 runs. The matrices become *more* singular as the amount of data increases. The sizes of the Hankel matrix correspond with Table 1.

selecting a length that is too short may result in a Hankel matrix whose approximate rank is strictly less than the number of states of the system. Our chosen lengths are included in Table 1. Automatically determining the proper Hankel size is an open question since the development of spectral approaches for PSRs [52, 12, 10], and remains an interesting question for future work.

The main parameter associated with learning PSRs is centered around the number of singular components to be used when computing the rank factorization of the Hankel matrix (Section 3.3). As mentioned in Section B.1, the main way to do this is by specifying a lower threshold on the empirical Hankel matrix lower rank approximation's reciprocal condition number. Empirically, we observe that empirical Hankel matrices tend to become more singular as the amount of data used to estimate them increases (Fig. 7). While any sufficiently small positive threshold may work with large amounts of data, in practice, larger thresholds will more quickly identify the number of states, at risk of omitting states. Of practical concern is the maximum number of singular values to compute to avoid computing an SVD of the *entire* Hankel matrix. The specified values for our experiments are shown in Table 1 under $1/\kappa$ and 'No. SVD,' respectively.

When recovering the observation distributions and partition-level transitions, we must specify thresholds to determine transition matrix full-rankness and a threshold that determines when observation distributions are similar (Section B.1). Inverting a near-singular matrix is highly undesirable when computing the matrices to joint-diagonalize in Eq. (15), so we specify conservatively high threshold on the smallest singular value on the smallest singular value σ_{min} of the transition matrix. As discussed in Appendix B.1, in practice, a threshold τ_{obs} is not required to compute a random block-diagonal rotation matrix R. However, τ_{obs} is still required to compute partition-level transition likelihoods errors reported in Figs. 5 and 6. We set a conservativitely high threshold τ_{obs} to merge observation distributions aggressively when plotting those figures.

All PO-UCT planners require specification of a upper-confidence bound (UCB) constant to balance exploiting current estimated action value and exploring new actions. For all experiments, we use a UCB constant of $c = 2$. Our planners also limits all searches to a depth of three, and performs a fixed 1000 simulations per planning step.

Table 1: Chosen parameters for each domain in planning experiments described in Section 5, Fig. 5. Up and down arrows indicate a upper or lower threshold, respectively. The tuple reported for maximum indexing sequence lengths is ordered: (rows, columns).

	$1/\kappa \downarrow$	$\sigma_{min} \uparrow$	$\mathcal{H}$ max. seq. len. $\uparrow$	No. SVD $\uparrow$	$\tau_{obs} \uparrow$
Tiger	0.34	0.1	(2, 1)	20	0.1
T-Maze	0.1	0.01	(2, 1)	20	0.1
Sense-Float-Reset (3 states)	0.1	0.1	(3, 2)	20	0.1
Sense-Float-Reset (4 states)	0.015	0.1	(4, 3)	20	0.5

C.2 Sensitivity Analysis and Wall-Clock Runtime Estimates

Additionally, we have included an analysis on the sensitivity of the truncated SVD step to both Hankel size and rank tolerance $1/\kappa$ (see Appendix C.1 for a description on parameters). As experiments results in Fig. 5 suggest, convergence of the number of states is a key step to the convergence of the overall algorithm.

Table 2 reports the number of estimated states against a variety of Hankel sizes and rank tolerances (and the rank of the Hankel matrix in limit of infinite data). Estimates of observation and transition likelihood error can be found in Tables 3 and 4 respectively. Runtimes can be found in Table 5. Maximum Hankel size was determined to be the largest to allow for a RAM allocation under 64Gb on eight cores allocated on an Intel Xeon Gold 6140 CPU on a shared cluster. We observe that while more aggressive rank thresholds may arrive at the correct estimate with less data, they may also lead to an underestimation of the number of states. Lower thresholds will more likely underestimate the number of states, and require more data before the correct estimate is reached. Furthermore, larger Hankel sizes are required to estimate POMDPs with larger states, which tend to have longer observability lengths (Appendix B.2). For example, a Hankel size that enumerates histories of length four and tests of length three cannot fully capture a 14-state T-Maze. As Hankel size increases, so do the acceptable thresholds to estimate the number of states. Transition and observation likelihood errors, however, appear to be less influenced by Hankel size. These results suggest that the largest possible Hankel accomodated by running time and memory should be used for ease of selection of the remaining algorithm parameters.

We have also taken runtime estimates of the runtime of each component of the learning algorithm and PO-UCT search for the results reported in Fig. 5, which we have included in Table 6. The algorithms have been implemented as unoptimized Python code running on two cores allocated from an Intel Xeon Gold 6140 CPU and 4Gb RAM on a shared cluster. We observe the most expensive part of the algorithm is the estimation of the Hankel matrix, because its length and width scales exponentially as we extend the maximum length of enumerated histories and tests (Appendix B.2). The remaining learning components of the algorithm can be highly vectorized, and when learning POMDPs of with

Table 2: Sensitivity analysis on the **estimated rank** of the Hankel matrix based chosen Hankel sizes and rank tolerances estimated from 10^7 interactions in T-Maze environments with varying numbers of states. Hankel size is represented in the maximum lengths of action-observation sequences used to index the row and columns of the Hankel matrix, respectively. Rank tolerance is specified as $1/\kappa$, as discussed in Appendix C.1. Results are reported up to two significant figures, with trailing zeros truncated for space. Hankel rank for the corresponding Hankel size in the limit of infinite data is included in the GT column. Results reported are mean and standard deviation of the number of estimated states, aggregated over 20 seeds. A value of 'nan' is reported when no full-rank actions were found.

	$1/\kappa$	GT	1e-01	1e-02	1e-03	1e-04	1e-05	1e-06
n states	$\mathcal{H}$ seq. len.							
4	(2, 1)	4	4 ± 0	4 ± 0	$6 \pm .74$	$13 \pm .73$	$14 \pm .46$	14 ± 0
	(3, 2)	4	4 ± 0	4 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	4	4 ± 0	4 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
6	(2, 1)	5	4 ± 0	5 ± 0	$7.9 \pm .94$	$16 \pm .73$	$18 \pm .4$	18 ± 0
	(3, 2)	6	5 ± 0	6 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	6	6 ± 0	6 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
8	(2, 1)	6	5 ± 0	6 ± 0	10 ± 1.1	$20 \pm .3$	20 ± 0	20 ± 0
	(3, 2)	8	6 ± 0	8 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	8	6 ± 0	8 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
10	(2, 1)	7	6 ± 0	7 ± 0	13 ± 1.4	20 ± 0	20 ± 0	20 ± 0
	(3, 2)	9	6 ± 0	8 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	10	7 ± 0	$10 \pm .3$	20 ± 0	20 ± 0	20 ± 0	20 ± 0
12	(2, 1)	8	7 ± 0	8 ± 0	$17 \pm .93$	20 ± 0	20 ± 0	20 ± 0
	(3, 2)	10	7 ± 0	9 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	12	8 ± 0	14 ± 1.2	20 ± 0	20 ± 0	20 ± 0	20 ± 0
14	(2, 1)	9	8 ± 0	9 ± 0	$20 \pm .57$	20 ± 0	20 ± 0	20 ± 0
	(3, 2)	11	8 ± 0	10 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0
	(4, 3)	13	9 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0	20 ± 0

Table 3: Sensitivity analysis on the **observation error** (in L_1 norm) associated with Table 2. Estimates are only taken when the number of estimated states is equivalent to the ground truth POMDP. For all experiments the full-rank transition threshold σ_{min} is set to 0.01. A value of 'nan' is reported when no full-rank actions were found. If no standard deviation is included, only one seed of twenty succeeded in finding a full-rank action.

	$1/\kappa$	1e-01	1e-02	1e-03	1e-04	1e-05	1e-06
n states	$\mathcal{H}$ seq. len.						
4	(2, 1)	0.09 ± 0.2	0.11 ± 0.28	nan	nan	nan	nan
	(3, 2)	0.031 ± 0.012	0.033 ± 0.015	nan	nan	nan	nan
	(4, 3)	0.026 ± 0.011	0.025 ± 0.012	nan	nan	nan	nan
6	(2, 1)	nan	nan	0.47	nan	nan	nan
	(3, 2)	nan	0.2 ± 0.45	nan	nan	nan	nan
	(4, 3)	0.063 ± 0.025	0.077 ± 0.054	nan	nan	nan	nan
8	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	0.38 ± 0.24	nan	nan	nan	nan
	(4, 3)	nan	0.59 ± 1.5	nan	nan	nan	nan
10	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	0.29 ± 0.13	nan	nan	nan	nan
12	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	1.7 ± 0.017	nan	nan	nan	nan
14	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	nan	nan	nan	nan	nan

Table 4: Sensitivity analysis on the **transition error** (in L_1 norm) associated with Table 2. Values are reported using the same estimation protocol as Table 3, except transition likelihoods were measured.

	$1/\kappa$	1e-01	1e-02	1e-03	1e-04	1e-05	1e-06
n states	$\mathcal{H}$ seq. len.						
4	(2, 1)	0.095 ± 0.32	0.067 ± 0.21	nan	nan	nan	nan
	(3, 2)	0.018 ± 0.015	0.028 ± 0.045	nan	nan	nan	nan
	(4, 3)	0.017 ± 0.022	0.011 ± 0.0077	nan	nan	nan	nan
6	(2, 1)	nan	nan	0.46	nan	nan	nan
	(3, 2)	nan	0.073 ± 0.13	nan	nan	nan	nan
	(4, 3)	0.021 ± 0.014	0.06 ± 0.091	nan	nan	nan	nan
8	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	0.033 ± 0.024	nan	nan	nan	nan
	(4, 3)	nan	0.13 ± 0.36	nan	nan	nan	nan
10	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	0.047 ± 0.038	nan	nan	nan	nan
12	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	0.23 ± 0.0087	nan	nan	nan	nan
14	(2, 1)	nan	nan	nan	nan	nan	nan
	(3, 2)	nan	nan	nan	nan	nan	nan
	(4, 3)	nan	nan	nan	nan	nan	nan

small numbers of states (fewer than four states), their runtimes are relatively fast.

C.3 Planning Performance of Different Sampling and Distribution Rounding Strategies

There are many ways to sample action-observation trajectories when deriving a UCT-based search algorithm for planning on POMDPs. As discussed by Silver and Veness [46], there are largely two approaches, which differ in how the latent state treated as the search propogates down a branch of the search tree.

1. Upon choosing an action, propogate the full belief state using a process update (e.g. multiplying by $\tilde{T}^a$ of Theorem 1). Compute the mixture observation distribution weighted by that belief state, from which we can sample the emitted observation.
2. Upon choosing an action, sample a *single* latent state (or observability partition, in the context of Theorem 1). Look up the observation distribution associated with the action and sampled state, and then sample the observation.

Silver and Veness [46, Lemma 2] prove that the observation distribution under these two sampling strategies are equivalent, so a UCT-based search will perform

Table 5: **Runtime estimates** for sensitivity analysis on the T-Maze environments shown in Table 2. All Hankel matrices were estimated at maximum size using sparse representations and then indexed to form smaller Hankel matrices, which is why there is little variation in runtime across T-Maze instances with varying numbers of states. All estimates are reported as mean and standard deviation over 20 seeds.

$\mathcal{H}$ seq. len.	n states	PSR (s)	POMDP (s)
$(2,1)$	4	0.19 ± 0.098	0.2 ± 0.099
	6	0.24 ± 0.12	0.25 ± 0.12
	8	0.25 ± 0.11	0.25 ± 0.11
	10	0.29 ± 0.12	0.29 ± 0.12
	12	0.33 ± 0.15	0.34 ± 0.16
	14	0.35 ± 0.13	0.36 ± 0.13
$(3,2)$	4	1.1 ± 0.41	1.1 ± 0.41
	6	1.6 ± 0.54	1.6 ± 0.54
	8	2.3 ± 0.77	2.3 ± 0.77
	10	3.1 ± 0.93	3.1 ± 0.93
	12	3.9 ± 1.3	3.9 ± 1.3
	14	6.1 ± 13	6.1 ± 13
$(4,3)$	4	77 ± 32	77 ± 32
	6	140 ± 33	140 ± 33
	8	250 ± 75	250 ± 75
	10	450 ± 130	450 ± 130
	12	740 ± 220	740 ± 220
	14	1.0e+3 ± 260	1.0e+3 ± 260

n states	$\mathcal{H}$ estim. time (s)
4	910 ± 210
6	930 ± 190
8	960 ± 220
10	980 ± 220
12	970 ± 220
14	920 ± 200

Table 6: Runtime estimates from Fig. 5. PSR and POMDP columns are the total estimated times for learning each model, respectively. The last column describes the average planning time per planning step. The entries report mean and standard deviations, in seconds, up to two significant figures. Hankel estimates are the total amount of time to evaluate Eq. 4 on 10^6 interactions.

	$\mathcal{H}$ estim. (s)	PSR (s)	POMDP (s)	EM (s)	Plan. (s/step)
Tiger	16 ± 8.6	0.013 ± .0061	0.015 ± 0.0068	3.7 ± 2.4	3.4 ± 1.2
T-Maze	18 ± 12	0.022 ± 0.041	0.024 ± 0.015	5.7 ± 4.2	4.1 ± 2
SFR (3 states)	27 ± 16	0.017 ± 0.011	0.019± 0.011	15 ± 10	3.5 ± 1.4
SFR (4 states)	83 ± 28	0.29 ± 0.14	0.29 ± 0.14	350 ± 250	4.7 ± 1.6

the same using either approach. They also argue the latter is more computationally efficient for systems with a large number of states.

UCT-based search algorithm may only use one or some variant of both approaches when planning with the models learned by the algorithms discussed in this paper. Because PSRs do not yield explicit transition likelihood estimates, we do not have the state distributions used to sample individual states for the second approach. The first approach, however, can still be applied. Given a PSR sufficient statistic m, we compute products $m \cdot M^{ao} \cdot m_\infty$ for the chosen action and all possible observations, yielding the observation likelihoods. The learned partition-level POMDPs may apply the first approach in the same way. Furthermore, the second approach may be applied to the learned partition-level POMDPs by sampling the next *observability partition*, rather than state. Given a partition-level belief $\tilde{b}$, we first compute the partition-level belief distribution by summing across across appropriate indices, and then sampling the current partition S. We can then compute the *conditional* observation distribution by first computing the *conditional* partition-level belief vector

$$\tilde{b}_S = \frac{\tilde{b} \otimes \mathbb{I}_S}{(\tilde{b} \otimes \mathbb{I}_S)^T \cdot \mathbf{1}}$$

where $\mathbb{I}_S$ is a vector with entries of value one for indices in partition S and zero otherwise, a is the selected action by the search algorithm, and $\otimes$ the element-wise product. The conditional observation distribution is then found by computing products $\tilde{b}_S \tilde{T}^a O^{ao}$ for all observations $o \in \mathcal{O}$, and projecting the distribution to deal with approximation error as handlied in the first approach. Verifying the correctness of this calculation is a straightforward extension of the proof of Theorem 1.

Our approach to handling rounding estimated likelihoods to proper probability distribution parameters is different across the two sampling strategies. When planning using the first sampling approach, we first compute the estimated observation distribution, project the distribution, and then sample. When planning with the second, we compute the estimated partition-level likelihoods, project the distribution, sample a *partition*, and then sample the appropriate observation. In practice, we do not observe an empirical difference between these sampling and rounding approaches for planning with rewards learned as observations.

C.4 Slower Convergence of Transition Likelihoods

A common approach to convergence analysis of tensor decomposition methods would first argue the convergences of the SVD of our Hankel matrix and *then* argue convergence of the eigendecompositions of the matrix discussed in Eq. (16) [36]. PSRs only depend on the convergence of the SVD, while our learned POMDPs depends on the convergence of both the SVD and eigendecomposition. In our reward-specification experiments (Fig. 6), accurate transitions are required to correctly assign reward to the desired goal state. The slow convergence of performance of the planner in the directional hallway environment

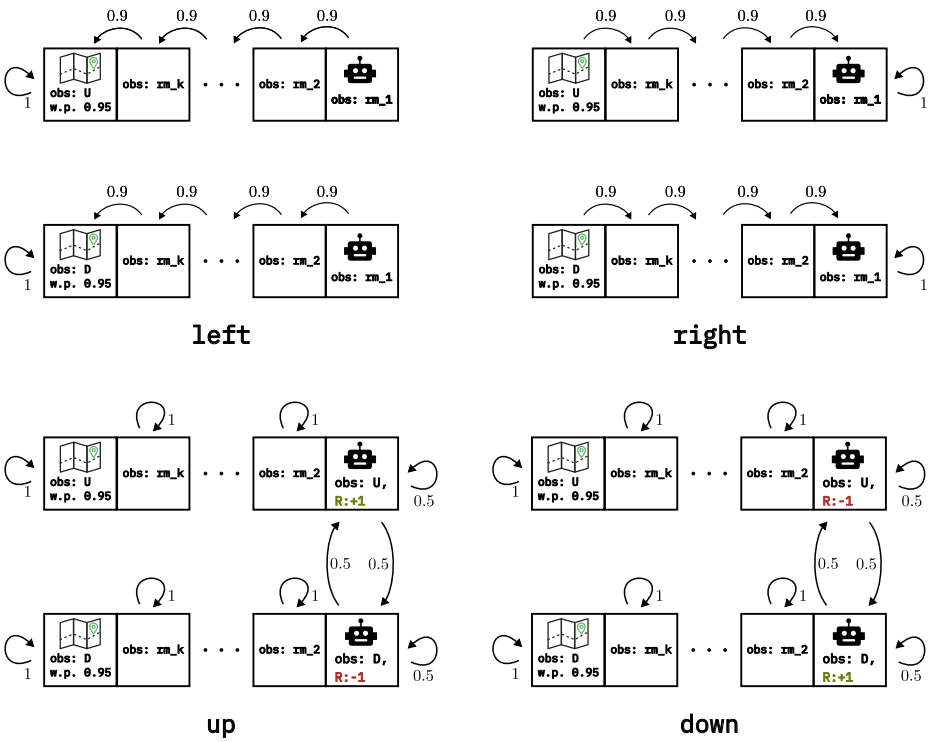

Fig. 8: T-Maze dynamics and observation distributions. Edges are labeled with transition probabilities, and nonzero rewards are emitted deterministically from annotated states (and rewards of zero from non-annotated states). Self-loop edges with probability less than 1 are omitted. Leftmost 'map' states in the top hallway emit a `U` with probability with probability 0.95 and `D` with probability 0.05 (and vice-versa for the bottom hallway). All other observations are deterministic.

suggests that more data is required to obtain accurate likelihood estimates of transition and diagonal observation matrices.

C.5 Experimental Domains

Here, we document any environment we have modified, or any novel environments we introduced in this work. We have used the original Tiger domain as described by [29], which we omit from our discussion below.

Sense-Float-Reset As discussed in Sec. 4, the transition dynamics and observation emissions of Sense-Float-Reset are the same as those of Float-Reset introduced by [32], but augment the system with a passive sensing action.

Transition dynamics. In an n state float-reset problem, the 'reset' state is typically denoted as s^0 and the remaining states $\{s^1, s^2, \ldots, s^{n-1}\}$. The `float`

action allows the system to translate to adjacent integer states (or loop at the ends):

$$\mathrm{P}(s_{t+1} = s^j | s_t = s^i, a_t = \texttt{float}) = \begin{cases} 0.5, & i = j = 0, (n-1) \text{ or } i = j \pm 1, \\ 0 & \text{otherwise} \end{cases}$$
$$\forall i, j \in \{0, \ldots, n-1\}.$$

The `reset` action deterministically sets the state to s^0, e.g.

$$\mathrm{P}(s_{t+1} = s^0 | s_i, a_t = \texttt{reset}) = 1 \quad \forall i \in \{0, \ldots, n-1\}.$$

The `sense` action does not change the state, e.g.

$$\mathrm{P}(s_{t+1} = s^i | s_t = s^i, a_t = \texttt{sense}).$$

Observation emissions. The `float` action only emits an observation of zero, e.g.

$$\mathrm{P}(o_t = \texttt{0} | s_t = s^i, a_t = \texttt{float}) = 1 \quad \forall i \in \{0, \ldots, n-1\}.$$

The `reset` and `sense` actions emit a `1` when s_t is in s^0 (the 'reset state'), and `0` otherwise:

$$\begin{aligned} &\mathrm{P}(o_t = \texttt{1} | s_t = s^i, a_t = \texttt{reset}) = \mathrm{P}(o_t = \texttt{1} | s_t = s^i, a_t = \texttt{sense}) \\ &\quad = \begin{cases} 1, & i = 0, \\ 0, & \text{otherwise}, \end{cases} \\ &\mathrm{P}(o_t = \texttt{0} | s_t = s^i, a_t = \texttt{reset}) = \mathrm{P}(o_t = \texttt{0} | s_t = s^i, a_t = \texttt{sense}) \\ &\quad = \begin{cases} 1, & i \in \{1, \ldots, n-1\}, \\ 0, & \text{otherwise}. \end{cases} \end{aligned}$$

Reward function. In all of our experiments, we specify a deterministic tabular reward of $+1$ when the system exists s^1, and emit a reward of 0 otherwise,

$$\mathrm{P}(r_t = r | s_t = s^i, a_t = a) = \begin{cases} 1, & r = +1, i = 1 \text{ or } r = 0, i \in \{0, 2, \ldots n-1\} \\ 0, & \text{otherwise}. \end{cases}$$
$$\forall a \in \{\texttt{float}, \texttt{reset}, \texttt{sense}\}.$$

T-Maze We present a version of T-Maze similar to the one described by [2]. Since we allow actions to determine observation emissions ([2] determine observations by states), our T-Maze POMDP has fewer states than their version. This environment is more easily explained pictorally than explicit probability expressions. See Fig. 8 for a depiction of transition dynamics and observation emissions. For the truncated T-Maze used for experiments in Section 5, Fig. 5, the number of room states was set to $k = 1$.

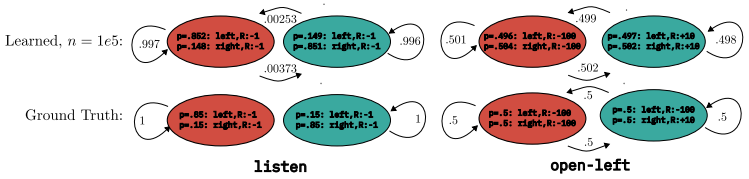

Fig. 9: A comparison of a learned instance of Tiger after 10^5 samples compared to the ground truth for the `listen` and `open-left` actions. For each action, nodes are annotated with their observation emission probabilities, and edges are annotated with their transition probabilities.

Noisy Hallways The transition dynamics common to *directional hallway* and *noisy hallway* can be found in Fig. 4. Across both domains, under the `stay` and `reset` actions, the environment will emit either `end-left` or `end-right` with probability 0.5. Furthermore, the left and the right states will emit `end-left` and `end-right`, respectively, with probability 0.8 under actions `left`, and will omit the incorrect observation (`end-right` from leftmost state, and vice-versa) with probability 0.2.

The two domains differ on the observation distribution of the middle state under the actions `left` and `right`. In the directional environment, under `left`, the observation `end-left` is emitted with probability 0.8, and the `end-right` emitted with probability 0.2. Similaritly, under the `right` action, the `end-right` observation is emitted with probability 0.8, and the `end-left` observation is emitted with probability 0.2. In the noisy environment, under both `left` and `right` actions, either `end-left` or `end-right` may be emitted with probability 0.5.

There is no reward function is given, since both of these experiments are used in the reward-specification experiments discussed in Sec. 5, Fig. 6. For the directional environment, the tuples that are assigned rewards are (`left`, `end-left`) and (`right`, `end-right`). For the noisy environment, the tuples assigned rewards are the elements of the set $\{\texttt{left}, \texttt{right}\} \times \{\texttt{end-left}, \texttt{end-right}\}$.

It is important to note that these domains are fully-recoverable by our algorithm, even though there are fewer observations than states. This is because all actions aside from `reset` are full-rank and that the observation *distributions* associated with these actions are distinct.

C.6 Example Output of Algorithm

Here, we include an example of the learned model after estimates of transition and diagonal observation matrices have nearly converged. In Tiger, where each state has a unique observation distribution, the learned model, as illustrated in Fig. 9, shows close agreement between the learned and ground-truth transition and diagonal observation matrices. These results confirm that by estimating the similarity transform, we can recover the true observation and transition models.