

Toward Learning POMDPs Beyond Full-Rank Actions and State Observability

Seiji Shaw¹, Travis Manderson¹, Chad Kessens², and Nicholas Roy¹

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 Azzadenesheli 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 $T_{ij}^a = P(s_{t+1} = s^j | s_t = s^i, a_t = a)$ 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 $O_{ii}^{ao} = P(o_t = o | s_t = s^i, a_t = a)$ 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 $a_t \sim \text{Unif}(\mathcal{A})$ 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

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

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 $\text{Obs}_{i,j}^a = P(o_t = o^j | s_t = s^i, a_t = a)$. To distinguish between the two, we refer to the collection \mathcal{Z} as *diagonal observation matrices* and the latter matrices $\{\text{Obs}^a : a \in \mathcal{A}\}$ 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 $hist = (a_1, o_1, \dots, a_t, o_t)$. 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:

$$\mathbf{Forw}_{hist,:} = b_0 \cdot O^{a_1, o_1} T^{a_1} \dots O^{a_t, o_t} T^{a_t}. \quad (1)$$

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 $test = (a_{t+1}, o_{t+1}, \dots, a_n, o_n)$, then a column of **Back** can also be computed similarly to the rows of **Forw**:

$$\mathbf{Back}_{:,test} = O^{a_{t+1}, o_{t+1}} T^{a_{t+1}} \dots O^{a_n, o_n} T^{a_n} \cdot \mathbf{1}, \quad (2)$$

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 $\mathbf{1}$ 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 $hist = (a_1, o_1, \dots, a_t, o_t)$ and a test $test = (a_{t+1}, o_{t+1}, \dots, a_n, o_n)$, 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.

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

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 $\mathcal{D}_n = (a_1, o_1, \dots, a_n, o_n)$, the matrix \mathcal{H} could be estimated by the *suffix-history approach*, taking frequency counts of subsequences of increasing lengths [52, 12]:

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

where $acts(hist \oplus test)$ 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 $a \sim \pi_{\text{exp}}(\mathcal{A})$, $\pi \in \Delta(\mathcal{A})^4$ (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 **Forw**·**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 $a \in \mathcal{A}, o \in \mathcal{O}$:

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

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

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

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, \dots, o_n under actions a_1, \dots, a_n by computing the product $P(o_1, \dots, o_n | a_1, \dots, a_n) = m_0^T M^{a_1 o_1} \dots M^{a_n o_n} m_\infty = b_0 P P^{-1} O^{a_1 o_1} T^{a_1} \dots O^{a_n o_n} T^{a_n} P^{-1} P \mathbf{1}$, 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 $\text{rank}(\mathcal{H}) = r$ 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 emit 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

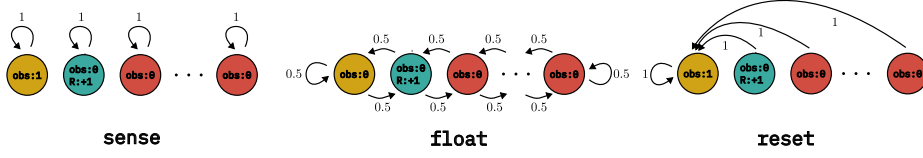


Fig. 1: Sense-Float-Reset. Edges are labeled with transition probabilities, and nodes are labeled with observations and reward received upon *leaving* the state. The reward of a state is zero unless specified otherwise. Node shade denotes observability partitions.

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 $S \in \mathcal{S}_\Pi$, states $s^i, s^j \in S$ if and only if $\text{Obs}_{i,:}^a = \text{Obs}_{j,:}^a$, for all $a \in \mathcal{A}_{full}$. We call \mathcal{S}_Π 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 $\mathcal{S}_\Pi \subset 2^{\mathcal{S}}$ 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

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

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

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

then

$$\sum_{s^i \in S} \tilde{b}_{\pi i} = \sum_{s^i \in S} b_{\pi i} \quad (11)$$

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

$$\tilde{b}_\infty = \mathbf{1} \quad (13)$$

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

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 $p_{succ}T + (1 - p_{succ})I$, 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. $p_{succ} \neq 1/2, 1$), 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 $a \in \mathcal{A}$:

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

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 $M^a \in \mathcal{M}_{full}$ and $o \in \mathcal{O}$ we compute

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

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

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

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 $\text{Unif}(\mathbb{S}^{|\mathcal{A}_{full}| \cdot |\mathcal{O}| - 1})$ 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 $O_{ii}^{ao} = O_{jj}^{ao}$ 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,

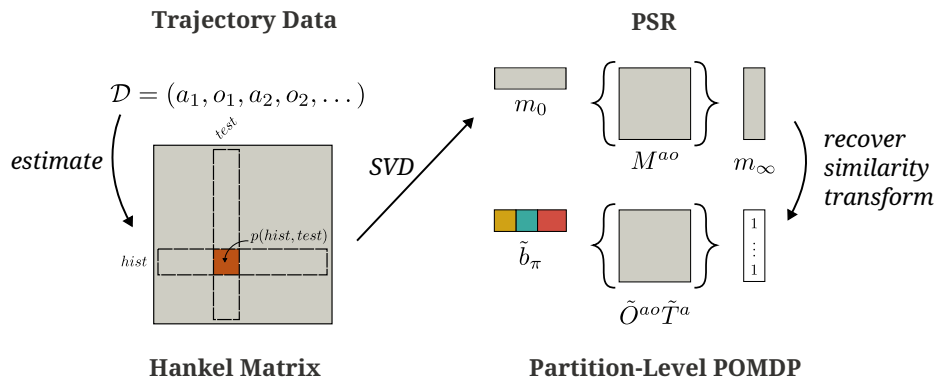


Fig. 3: An illustration of the entire algorithm pipeline described from Sec. 3.1-4.

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. $P m_0 = P'^{-1} P \mathbf{1}$. Intuitively, by applying $\text{diag}(P m_0)$ 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 $\text{diag}(R P'^{-1} m_\infty) R P'^{-1}$ 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

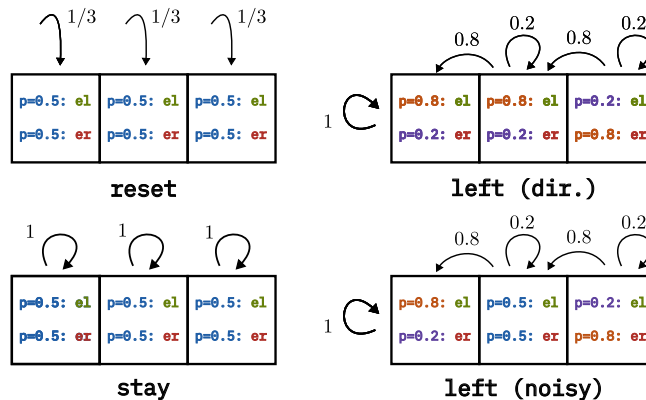


Fig. 4: The directional and noisy hallway domains. The observations e_l and e_r represent ‘end-right’ and ‘end-left’ respectively. Transitions are shared and observation distributions differ in the middle state in the **left** and **right** actions. The **right** action is the mirror image of **left**, 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,

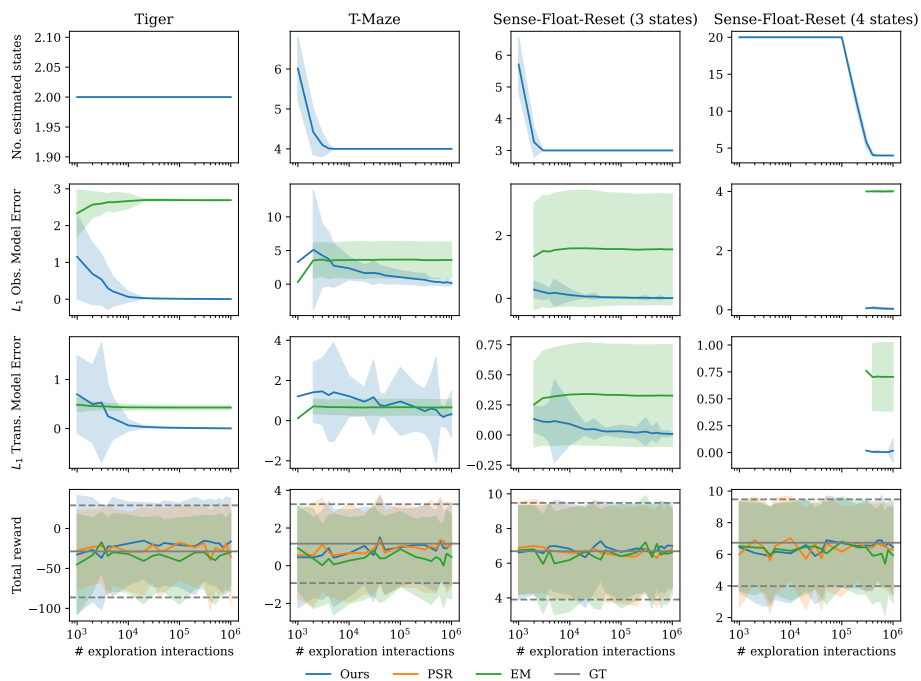


Fig. 5: Error bars represent standard deviation over 100 seeds. The y-axis is scaled to make convergence visible. **Row 1:** Estimated number of states. **Row 2:** Obs. matrix error relative to ground truth. **Row 3:** Trans. matrix error. This error is only measurable once the estimated number of states 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

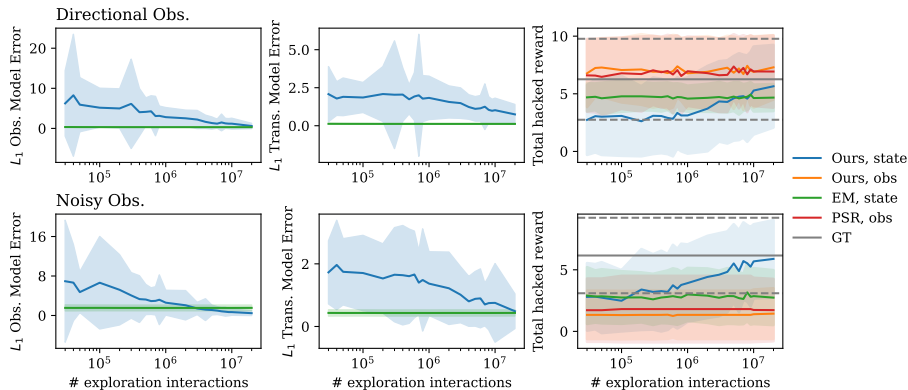


Fig. 6: The agent receives +1 reward for each timestep in the designated goal state (middle of hallway). ‘Obs’ refers to assigning rewards to action-observation pairs, whereas ‘state’ refers to assigning rewards to states. Error bars report standard deviation over 100 seeds. ‘Hacked reward’ refers to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

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

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 $\mathcal{H} = \lim_{n \rightarrow \infty} \hat{\mathcal{H}}$, 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-Maximization 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.: Per-Sim: 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 Re-planning 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)