Discrete Approximate Information States in Partially Observable Environments

by

Lujie Yang

B.S., University of California Berkeley (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Author ....
Department of Electrical Engineering and Computer Science
January 23, 2023

Certified by ....
Russell L. Tedrake
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Theses

Discrete Approximate Information States in Partially Observable Environments

by

Lujie Yang

Submitted to the Department of Electrical Engineering and Computer Science
on January 23, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The notion of approximate information states (AIS) was introduced in [31] as a methodology for learning task-relevant state representations for control in partially observable systems. They proposed particular learning objectives which attempt to reconstruct the cost and next state and provide a bound on the suboptimality of the closed-loop performance, but it is unclear whether these bounds are tight or actually lead to good performance in practice. Here we study this methodology by examining the special case of *discrete* approximate information states (DAIS). In this setting, we can solve for the globally optimal policy using value iteration for the DAIS model, allowing us to disambiguate the performance of the AIS objective from the policy search. Going further, for small problems with finite information states, we reformulate the DAIS learning problem as a novel mixed-integer program (MIP) and solve it to its global optimum; in the infinite information states case, we introduce clustering-based and end-to-end gradient-based optimization methods for minimizing the DAIS construction loss. We study DAIS in three partially observable environments and find that the AIS objective offers relatively loose bounds for guaranteeing monotonic performance improvement and is sufficient but not necessary for implementing optimal controllers. DAIS may even prove useful in practice by itself or as part of mixed discrete- and continuous-state representations, due to its ability to represent logical state, to its potential interpretabilty, and to the availability of these stronger algorithms.

Thesis Supervisor: Russell L. Tedrake
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I am extremely grateful to my advisor Russ, whose incredible passion and stamina for solving challenging robotic tasks motivate me in my research journey all the time. Russ has always been patient and encouraging while holding our work to exceptionally high standards. His deep insights into valuable research questions and remarkable knowledge base ranging from practical details to theoretical hierarchy create an ideal environment for us to explore and thrive.

I appreciate tremendous help from my collaborator Kaiqing, who patiently went through all the technical details with me and guided me to think as a more professional researcher. I would like to thank my coauthors Alex, whose mathematical rigor always polishes our work into more precise statements, and Yunzhu, who taught me a lot about the “black magic" for neural networks. I would like to acknowledge Hongkai, who has always kindly listened to my confusion and helped me think about concrete formulations. I also appreciate Jack, Terry and other RLG members for valuable discussions and feedback on my projects. It has been a great honor working in such a diverse and brilliant research group during the last two years.

Contents

1 Introduction **11**

1.1 Motivation 11

1.2 Contributions 13

1.3 Organization 13

2 Related Works **15**

2.1 Representation Based on Observation Reconstruction/Prediction . . . 15

2.2 POMDP Solvers 16

2.3 Output Feedback Control 16

2.4 State Aggregation 16

3 Background **19**

3.1 Partially Observable Markov Decision Process 19

3.2 Approximate Information State 20

4 Problem Formulation **23**

4.1 Finite Belief Space 23

4.2 Infinite Belief Space 23

4.2.1 Continuous-State POMDPs 24

4.2.2 Visual-Feedback Control Tasks 25

5 Discrete Approximate Information States **27**

5.1 Finite Belief Space 29

5.1.1 Reformulating Bilinear Optimization Problem 30

5.2 Infinite Belief Space 32
5.2.1 Gradient-Based Optimization 32
5.2.2 Clustering-Based Optimization 33
5.3 Planning 34
5.4 Results 34
5.4.1 CheeseMaze 34
5.4.2 Corridor Navigation 37
5.4.3 Visual-Feedback Control for Object Pile Manipulation 39

6 Conclusion **43**

List of Figures

1-1 Examples of tasks DAIS deals with. 12

5-1 DAIS learning framework. The red path shows that the current belief b_t propagates to the next time step b_{t+1} under Bayes rule C^{a_t} with action a_t, which is then discretized to the DAIS under transformation D at time $t+1$ as Z_{t+1}; the blue path demonstrates that the current belief b_t is first discretized to Z_t and then propagates to Z_{t+1} under the learned transition B^{a_t}. We aim to minimize the discrepancy between the probability distribution of Z_{t+1} obtained by the two paths as well as the difference between the reward predicted by b_t (i.e. $\hat{r}^{a_t}$) and Z_t (i.e. R_t). 28

5-2 (top) DAIS loss, task performance from empirical rollouts and value function approximation error vs maximum DAIS dimension in Cheese-Maze example. (bottom) DAIS loss and task performance vs exact DAIS dimension for Corridor Navigation with 10 random seeds. DAIS achieves higher return, more robust performance and much smaller runtime compared to CPBVI. 37

5-3 DAIS performance in object pile manipulation task: the robot manipulator is required to push the object pieces into the blue region. . . . 41

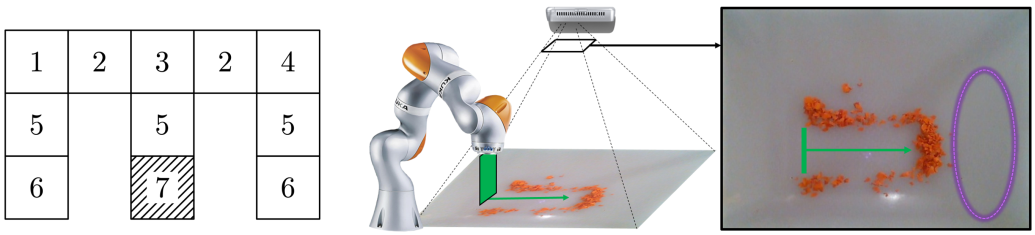
Chapter 1

Introduction

1.1 Motivation

In most autonomous control applications, the agent (or controller) only has access to partial observations of the system state [27, 9, 16, 32]. Common examples include robot navigation [27, 9] and robotic manipulation [16, 32]. The key to planning and control in such partially observable systems is constructing a *state representation*: a function of the partial observations through which we can predict future performance of future control actions. There is ever-growing literature on representation learning for control in partially observable systems, ranging from the classic state estimation (filtering) in linear systems [19], to the deep learning-based approaches of learning for control from pixels [14, 21, 36, 10, 34, 17].

Many of these recent approaches to representation learning for control from pixels are built upon *observation reconstruction/prediction*. In particular, they encode the history of observations (high-dimensional images) into lower-dimensional vectors to reconstruct and predict future observations [14, 21, 36]. Notably, these approaches are *task-agnostic*: the constructed representations are designed to recover all the information in the observations, including information irrelevant to the downstream control tasks. Such irrelevant information may easily distract the control and planning modules [38]. Moreover, no theoretical guarantees for the control performance were established for these types of representations, and the observations (images) are



(a) CheeseMaze (b) Object Pile Manipulation

Figure 1-1: Examples of tasks DAIS deals with.

usually high-dimensional and challenging to reconstruct/predict.

On the other hand, when modeled by the framework of partially observable Markov decision processes (POMDPs), the state representation that is sufficient for performance evaluation and optimal control is known to be different from those that are necessary for predicting the observations. Specifically, it is known that the belief state (i.e. the posterior belief of the unobserved state given the action-observation history) is a sufficient statistic for POMDPs, on which the optimal policy can be defined and identified via dynamic programming [2]. In fact, the belief state belongs to a more general notion of *information state* [20, 31] – a function of history which is sufficient to: 1) compute the expected reward; 2) predict the next information state. In [31], the authors showed that these two conditions are sufficient for performance evaluation in POMDPs. More importantly, it is also shown in [31] that any state representation that satisfies these two conditions approximately with uniform bounds over all possible observation/action histories, can be used to construct a Markov Decision Process (MDP) whose value function is a pointwise good approximation of the original POMDP therefore inducing a policy with bounded suboptimality. In other words, the two aforementioned properties provide rigorous metrics for the quality of a state representation of a POMDP based on its relevance to downstream optimal control. Such representations can thus be viewed as being *task-relevant*.

In this work, we study a *discrete* approximate information state (DAIS) representation. Specifically, we aim to discover the possible discrete nature of the approximate information state (AIS) in many structured POMDPs, which can potentially represent logical state and improve the interpretability. Moreover, a discrete AIS enables the use of optimal planning methods, e.g., value iteration, to solve the approximate model efficiently. Finally, constructing a discrete AIS facilitates the direct use of the two aforementioned conditions in training, without resorting to the surrogate conditions given in [31], which also requires predicting the potentially high-dimensional observations.

1.2 Contributions

The contributions of this thesis are summarized as follows. First, we present a framework to construct DAIS without observation prediction, for POMDPs with both finite- and infinite-cardinality belief states. Second, for the finite belief space POMDPs, we propose a mixed integer programming (MIP)-based formulation for constructing the optimal state representation, followed by a novel reformulation technique that yields a globally optimal solution. By solving small problems to optimality, we are able to study the gap between the DAIS loss bounds and the task performance. For the infinite belief space case, we develop both clustering-based and gradient-based methods and investigate the non-convex DAIS objective independently from the policy. We find that although the original AIS bound in [31] can be relatively loose for guaranteeing monotonic performance improvement, discrete model representations solved with exact value iteration can still yield optimal (or close to optimal in the infinite-belief case) control strategies. Third, we evaluate the effectiveness of DAIS on three benchmark partially observable environments, including a visual-feedback object pile manipulation task in robotics. We also demonstrate the interpretability of DAIS in some examples, and show the numerical advantages of planning over DAIS, compared to existing continuous-space POMDP solvers, e.g., [27].

1.3 Organization

The thesis is organized as follows: Chapter 2 reviews the related work on representation learning for controls in partially observable systems. Chapter 3 introduces the mathematical framework of approximate information states. Chapter 4 presents the formulation of partially observable markov decision process with both finite and infinite belief space. Chapter 5 provides tractable optimization programs to find discrete approximate information states. Chapter 6 concludes with a discussion of our work.

Chapter 2

Related Works

Besides the most relevant work [31] on AIS discussed above (and will be discussed more in Chapter 3), the other related works are summarized as follows.

2.1 Representation Based on Observation Reconstruction/Prediction

Faced with high-dimensional visual input, model-based reinforcement learning methods typically focus on reconstructing or predicting observations [37, 17, 14] to learn the underlying model for optimal planning. [34] employs variational autoencoders to learn a latent state with locally linear dynamics by accurately reconstructing the image at the next time step. [14] learns a latent space with both deterministic and stochastic dynamics by training multi-step predictions of observations and rewards. [38] aims to learn invariant representations without reconstruction, which has the closest motivation to ours. However, the framework was focused on the fully-observable settings of MDPs.

2.2 POMDP Solvers

Most exact POMDP algorithms utilize dynamic programming to compute the piecewise linear value function for optimal decision making under partial observability. Such exact compuation often suffers from the exponential growth of value function calculation. Various POMDP solvers like point-based value iteration [26, 30] and incremental pruning [5] have been proposed to avoid this major difficulty via convex approximations. However, most of the solvers are restricted to discrete state spaces and require extensive iterations to update the value function. [27] deals with continuous POMDP but the algorithm is fairly slow to train and is sensitive to model parameters (as observed in Section 5.4).

2.3 Output Feedback Control

In the controls community, there has been extensive literature on output feedback control that guarantees stability and robustness of the closed-loop systems where the states are unavailable for the controller design. Static output feedback [33, 4, 11] parameterizes the controller only as a function of the output while dynamic output feedback [1, 28] keeps an internal state for the controller update, which attempts to summarize the history of observations and controls, and resembles the notion of information states. However, most of the methods with theoretical guarantees only handle linear systems where the observations are linear functions of the states and fail under complicated, high-dimensional observations such as images.

2.4 State Aggregation

There are a number of works on state discretization [3, 24] and state aggregation [35] in MDPs. In particular, Givan et al. [12] propose to aggregate MDPs using bisimulation, the strictest partitioning form for preserving most properties. Ferns et al. [8] soften the exact equivalence requirement in bismulation using bisimulation metrics, presenting state aggregation techniques for MDPs which combine "behaviorally similar" states given the distance between their rewards and state distributions. Castro et al. [6] extend the notion of bisimulation metrics to POMDPs. However, they do not provide viable algorithms for computing equivalence and aggregation in belief space. In this work, we formulate tractable optimization programs to learn DAIS as an effective discretization of the belief space.

Chapter 3

Background

In this chapter, we provide the background for understanding the DAIS framework. We start with describing the POMDP and then introduce the definition of approximate information state, originated from [31].

3.1 Partially Observable Markov Decision Process

A POMDP is formally defined as a tuple $\langle \mathrm{S}, \mathrm{A}, T, r, \Omega, O, \gamma \rangle$, where S is the set of the states of the world, A is a set of actions that the agent can execute, T is the stochastic transition function $T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$, $r(s, a)$ is the reward function, Ω is a set of possible observations, O is an observation model with $O(s, o) = P(o_t = o | s_{t+1} = s)$ and $\gamma \in [0, 1)$ is the discount factor. The history until time t, denoted by H_t, is the summary of the past observations and actions, i.e., $H_t = (o_{1:t-1}, a_{1:t-1})$. Another important notion for POMDP is the belief state b_t, which is the posterior probability distribution of the unobserved states at time t. The belief state summarizes the previous experience until time t and assumes the Bayesian update:

$$\begin{aligned} b_{t+1}(s') &= f(b_t, a_t, o_t) \\ &= \frac{P(o_t | s_{t+1} = s') \sum_s P(s_{t+1} = s' | s_t = s, a_t) b_t(s)}{P(o_t | b_t, a_t)}, \end{aligned} \tag{3.1}$$

where

$$P(o_t|b_t, a_t) = \sum_{s'} P(o_t|s_{t+1} = s') \cdot \sum_{s} P(s_{t+1} = s'|s_t = s, a_t) b_t(s). \tag{3.2}$$

The belief update can be used to form the "belief MDP" for planning by marginalizing over the future observations:

$$P(b_{t+1} = b|b_t, a_t) = \sum_{o} P(o_t = o|b_t, a_t) \mathbb{1}(b = f(b_t, a_t, o)). \tag{3.3}$$

3.2 Approximate Information State

Let (ϵ, δ) be positive real numbers, $(\mathrm{X}, \mathcal{G})$ be a measurable space, d denote a probability metric between two probability distributions $\mu, \nu \in \Delta(\mathrm{X})$ (the space of probability measures on X) such as the Wasserstein distance or the Total Variation [1] metrics. An approximate information state $\{Z_t\}_{t=1}^{T}$ is generated by a history compression function $\{\sigma_t : \mathrm{H_t} \to \mathrm{Z}\}_{t\geq 1}$, Markovian update kernel $\hat{P} : \mathrm{Z} \times \mathrm{A} \to \Delta(\mathrm{Z})$ and reward prediction function $\hat{r} : \mathrm{Z} \times \mathrm{A} \to \mathbb{R}$ where $Z_t = \sigma_t(H_t)$ and the following properties are satisfied for any t, any realization h_t of H_t, and any action a_t of $A_t \in \mathrm{A}$:

(AP1) Sufficient to predict the reward $R_t = r(s_t, a_t)$ approximately:

$$|\,\mathbb{E}[R_t|H_t = h_t, A_t = a_t] - \hat{r}(z_t, a_t)| \leq \epsilon.$$

(AP2) Sufficient to predict its Markovian transition approximately: for any Borel subset Y of Z, define $\mu_t(\mathrm{Y}) \coloneqq P(Z_{t+1} \in \mathrm{Y}|H_t = h_t, A_t = a_t)$, $\nu_t(\mathrm{Y}) \coloneqq \hat{P}(Z_{t+1} \in \mathrm{Y}|z_t, a_t)$,

$$d(\mu_t, \nu_t) \leq \delta.$$

In general, the condition **(AP2)** can be abstract to enforce. [31] has thus proposed the following two surrogate conditions that imply **(AP2)**, which might be easier to

[1] The total variation is given by $TV(\mu, \nu) = \frac{1}{2} \sum_x |\nu(x) - \mu(x)|$.

enforce.

(AP2'a) Evolves deterministically like a state: there exists a measurable update function $\phi : \mathrm{Z} \times \Omega \times \mathrm{A}$ such that

$$z_{t+1} = \phi(z_t, o_t, a_t).$$

(AP2'b) Sufficient to predict future observations approximately: for any Borel subset Y of Ω, define $\mu_t^o(\mathrm{Y}) := P(O_t \in \mathrm{Y} | H_t = h_t, A_t = a_t), \nu_t^o(\mathrm{Y}) := \hat{P}^o(O_t \in \mathrm{Y} | z_t, a_t)$, then

$$d(\mu_t^o, \nu_t^o) \leq \delta^o.$$

We denote the true value function of the history by $V(h_t)$ and the approximation obtained from AIS with dynamic programming by $\hat{V}(z_t)$, then we have the following bound on the value function approximation error [31, Theorem 9]:

$$|V(h_t) - \hat{V}(z_t)| \leq \alpha, \quad \text{with} \quad \alpha = \frac{\epsilon + \gamma \delta \rho_d(\hat{V})}{1 - \gamma},$$

where ρ_d is associated with the chosen probability metric related to the underlying extremization definition of that metric (see [31, Definition 6]). For f defined over a discrete set we have that for the Wasserstein distance $\rho_d(f) = ||f||_{\mathrm{Lip}}$, the Lipschitz semi-norm, and for Total Variation $\rho_d(f) = \frac{\max(f) - \min(f)}{2}$.

The power of AIS is that, by enforcing the conditions approximately with some relaxation error (ϵ, δ), the value function of the AIS model, i.e., $\langle \mathrm{Z}, \mathrm{A}, \hat{P}, \hat{r} \rangle$, is also pointwise close to the actual value function of the POMDP, up to an error that can be bounded linearly by (ϵ, δ). This provides a principled way to design state representations for control in POMDPs, with provable suboptimality guarantees.

In many robotics applications, the observations, i.e., images, are of high dimensions and can be challenging to reconstruct and predict (i.e. to enforce **(AP2'b)**). Hence, we propose to only use **(AP2)**, which becomes tangible and more tractable under the discrete AIS framework.

Chapter 4

Problem Formulation

We now present several common types of POMDPs that we aim to address in the ensuing chapters.

4.1 Finite Belief Space

Some basic POMDP examples with finite state-action and observation spaces also by nature have deterministic transition and observation models. In such models, the set of belief states, i.e., the exact information state, has finite cardinality. In this case, it is thus sensible to design *discrete* AIS. We take the CheeseMaze [23] as an example.

Example: (CheeseMaze) The maze environment consists of 11 states (grid cells) and 7 observations (numbers on the grid cells) as shown in Fig. 1-1. An agent in the maze desires to reach the goal state (shaded cell) where the cheese lands. Its movement in all four directions (north, south, east and west) and the observation functions are deterministic (e.g. $P(o_t = o | s_{t+1} = \text{bottom left cell}) = \mathbb{1}(o = 6)$).

4.2 Infinite Belief Space

In general, with stochastic transition dynamics and observation models, there are infinitely many reachable belief states b_t starting from an initial belief b_0. We propose to discretize the infinite-cardinality belief state space using the approximate information state conditions in Sec. 3.2. In particular, we focus on two such settings that are common in robotics applications: continuous-state space POMDPs, and vision-feedback control tasks.

4.2.1 Continuous-State POMDPs

Most existing algorithms for solving model-based POMDPs focus on discrete states while many real-world applications, such as robot navigation and manipulation, are naturally represented using continuous states. Note that in this case, the belief state becomes continuous and has infinite cardinality. We consider the same class of continuous POMDPs as studied in [27]. The dynamics are given by a Gaussian model $P(\cdot|s_t, a_t) = \phi(s_t + f(a_t), \Sigma^{a_t})$, where ϕ denotes a Gaussian distribution with mean $s_t + f(a_t)$ and covariance Σ^{a_t}. The reward $r_{a_t}(s_t)$ is modeled by a linear combination of Gaussian distributions $r_{a_t}(s_t) = \sum_{i=1}^{N_r} w_i^r \phi_i(s_t|m_i^{a_t}, \Sigma_i^{a_t})$, where w_i^r are weights, ϕ_i are Gaussian distributions, $m_i^{a_t}$ and $\Sigma_i^{a_t}$ are the mean and covariance, and N_r is a predefined number. The observation model $P(o_t|s_{t+1})$ is characterized by a Gaussian mixture model and by assuming uniform $P(s)$ and sampling N_o observation/state pairs:

$$P(o, s) = \sum_{i=1}^{N_o} w_i^o \phi_i(s|s_i^o, \Sigma_i^o),$$

$$P(o_t = o|s_{t+1} = s) = \frac{P(o, s)}{P(s)} \propto \sum_{i=1}^{N_o} w_i^o \phi_i(s|s_i^o, \Sigma_i^o).$$

The belief state is continuously valued and can be written as a Gaussian mixture:

$$b_t(s) = \sum_{j=1}^{N} w_j \phi_j(s|s_j, \Sigma_j), \tag{4.1}$$

where N is the number of Gaussian components, the weights $w_j > 0$ and $\sum_{j=1}^{N} w_j = 1$. Such a representation of belief states is a natural consequence of linear-Gaussian dynamics and Gaussian mixture observation functions because the belief update can then be computed in closed form with all the Gaussian based functions. During the rollouts, the number of Gaussian components needed to represent the belief state keeps increasing because the Bayesian update involves summing and multiplying Gaussian components of all actions and observations. Following [27], we employ the hierarchical clustering method described in [13] to compress the growing Gaussian mixture to one with a fixed number of components.

4.2.2 Visual-Feedback Control Tasks

We are interested in visual feedback manipulation tasks with quasi-static dynamics, where the observed images can serve as sufficient statistics of history. Essentially, the high-dimensional raw pixel images with reduced resolution can be viewed as $\boldsymbol{b}_t$, which are mapped into a low-dimensional discrete representations Z_t for reward and transition model prediction. We believe that image reconstruction or observation prediction is not necessary because the low-dimensional DAIS can capture the most essential information for the manipulation tasks.

Chapter 5

Discrete Approximate Information States

We aim to learn a set of discrete approximate information states where Z is a finite set and $|Z| = n_z$ (n_z is a positive integer). The discrete approximate information states can be encoded as one-hot vectors Z_t and their categorical distributions can be represented using vectors $\bar{z}_t \in \Delta(Z)$. For general AIS, it is only tractable to sample from the probability distribution μ_t as closed-form computation for the infinite-cardinality representation is prohibitively difficult. Meanwhile with *discrete* AIS, we can calculate the distribution $P(Z_{t+1}|h_t, a_t)$ exactly using Bayes rule and belief discretization. Moreover, instead of having to minimize the surrogate loss from samples as in the original AIS framework [31], it is straightforward to encode probability distributions of discrete variables in vectors and measure their distance. Starting from the current belief, we can obtain the probability distribution $\bar{z}_{t+1}$ of the DAIS at the next time step following the two paths depicted in Fig. 5-1. The red path shows that we can update the current belief b_t to b_{t+1} using Bayes rule and subsequently discretize b_{t+1} to Z_{t+1} using the discretization map D:

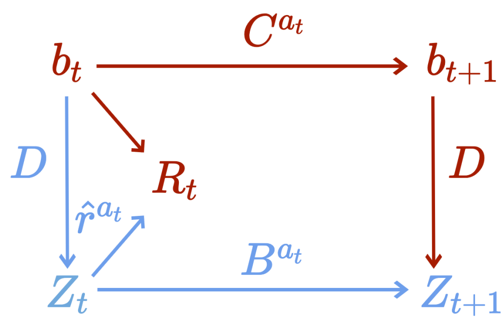


Figure 5-1: DAIS learning framework. The red path shows that the current belief b_t propagates to the next time step b_{t+1} under Bayes rule C^{a_t} with action a_t, which is then discretized to the DAIS under transformation D at time t+1 as Z_{t+1}; the blue path demonstrates that the current belief b_t is first discretized to Z_t and then propagates to Z_{t+1} under the learned transition B^{a_t}. We aim to minimize the discrepancy between the probability distribution of Z_{t+1} obtained by the two paths as well as the difference between the reward predicted by b_t (i.e. $\hat{r}^{a_t}$) and Z_t (i.e. R_t).

$$\begin{aligned} P(Z_{t+1}|h_t, a_t) &= P(Z_{t+1}|b_t, a_t) \\ &= \sum_b P(Z_{t+1}|b_{t+1} = b)P(b_{t+1} = b|b_t, a_t) \\ &= \sum_b \mathbb{1}(D(b) = Z_{t+1})P(b_{t+1} = b|b_t, a_t), \end{aligned} \tag{5.1}$$

where Z_{t+1} only comes from the discretization of the next time step belief b_{t+1} and is conditionally independent of b_t and a_t. Following the red path, we can also define the categorical distributions $\bar{z}_{t+1}$ as

$$\bar{z}_{t+1} = \begin{bmatrix} P(Z_{t+1} = z_1|b_t, a_t) \\ \vdots \\ P(Z_{t+1} = z_{n_z}|b_t, a_t) \end{bmatrix} = \sum_b P(b_{t+1} = b|b_t, a_t)D(b). \tag{5.2}$$

In general, D is a function that maps belief states, which are generally real-valued functions, to a finite set of categorical variables; in a discrete POMDP with deterministic dynamics and observations, D becomes a projection matrix that projects a large set of belief states down to a much smaller set of DAIS.

Meanwhile, the blue path shows that Z_{t+1} can also be obtained by first mapping b_t to Z_t and then propagating Z_t to the next time step under the learned transition matrix $B^a = [B^a_{ij}]_{i,j\in[n_z]}$, where $B^a_{ij} = \hat{P}(Z_{t+1} = z_i | Z_t = z_j, a_t = a)$:

$$\begin{aligned} P(Z_{t+1}|b_t, a_t) &= \sum_z P(Z_{t+1}|Z_t = z, a_t) P(Z_t = z|b_t) \\ &= \sum_z P(Z_{t+1}|Z_t = z, a_t) \mathbb{1}(D(b_t) = z), \end{aligned} \tag{5.3}$$

$$\bar{z}'_{t+1} = \hat{P}(Z_{t+1}|Z_t, a_t) D(b_t). \tag{5.4}$$

We aim to match the probability distribution of the next step DAIS $\bar{z}_{t+1}$ and $\bar{z}'_{t+1}$ obtained by the two procedures as well as the reward predicted by both the belief and DAIS. This framework explicitly avoids predicting observations and is beneficial when the output is high-dimensional (which is common in robotics applications). With the tabular "DAIS MDP", we can run value iteration to obtain the optimal planning policy for the approximate model.

In the following sections, we first describe how to formulate the finite-belief DAIS learning problem as an MIP and then extend it to the infinite-belief case with gradient-based and clustering-based optimization schemes.

5.1 Finite Belief Space

In discrete POMDPs with deterministic dynamics, the number of finite beliefs n_b is bounded. We would like to use a much smaller number ($n_z < n_b$) of DAIS to represent the task-relevant information of the belief optimally, i.e., minimizing the loss that enforces the AIS conditions **(AP1)** and **(AP2)**. Due to the discreteness of the belief space, we can describe each belief state b_t as a one-hot vector $\bar{b}_k$ (where the k-th entry is 1), write the belief update as a matrix multiplication and formulate the DAIS learning as a mixed-integer program:

$$\begin{aligned}
\min_{\{B^a\},D,\{\hat{r}^a\}} \quad & \sum_a \sum_{k=1}^{n_b} |r_k^a - \hat{r}^a D \bar{b}_k|^2 + \|B^a D \bar{b}_k - D C^a \bar{b}_k\|^2 \\
\text{s.t.} \quad & \mathbf{1}^T D = \mathbf{1}^T \\
& \mathbf{1}^T B^a = \mathbf{1}^T, \quad \forall a \\
& D_{ij} \in \{0,1\}, \quad \forall i,j \\
& B_{ij}^a \geq 0, \quad \forall i,j,a,
\end{aligned} \tag{5.5}$$

where we denote the belief MDP transition probability matrix by $C^a = [C_{ij}^a]_{i,j\in[n_b]}$ with $C_{ij}^a = P(b_{t+1} = \bar{b}_i | b_t = \bar{b}_j, a_t = a)$, the DAIS transition probability matrix by B^a. The projection matrix $D \in \{0,1\}^{n_z \times n_b}$ can only take entries 0 and 1, and has exactly a single 1 in each column because $\bar{b}_k$'s are transformed into one-hot encodings. $r_k^a = \mathbb{E}[R_t | b_t = \bar{b}_k, a_t = a]$ and $\hat{r}^a = [\hat{r}(z_1, a), \cdots, \hat{r}(z_{n_z}, a)]$ is the reward estimation vector with action a for all z. $\mathbf{1}$ denotes an all-one vector. The two terms in the objective enforce **(AP1)** and **(AP2)** respectively.

5.1.1 Reformulating Bilinear Optimization Problem

Notice that the optimization objective is bilinear in B^a and D as well as $\hat{r}^a$ and D. Such bilinear objectives are not mixed-integer convex. To make the optimization problem amenable to numerical computation, we use change of variables $Q^a = B^a D, \bar{r}^a = \hat{r}^a D$ and introduce binary auxiliary variables $\{t_{j_1 j_2}\}_{j_1,j_2}$ to reformulate the optimization problem exactly:

$$\begin{aligned}
\min_{\{Q^a\},D,\{\bar{r}^a\},\{t_{j_1 j_2}\}} \quad & \sum_a \sum_{k=1}^{n_b} |r_k^a - \bar{r}^a \bar{b}_k|^2 + \|Q^a \bar{b}_k - DC^a \bar{b}_k\|^2 \\
\text{s.t.} \quad & D_{ij} \in \{0,1\}, \quad \forall i,j \ \text{ and } \ \mathbf{1}^T D = \mathbf{1}^T \\
& Q_{ij}^a \geq 0, \quad \forall i,j,a \ \text{ and } \ \mathbf{1}^T Q^a = \mathbf{1}^T, \quad \forall a \\
& t_{j_1 j_2} \in \{0,1\}, \quad \forall j_1 \in [n_z], j_2 \in [n_z], j_1 < j_2 \\
& t_{j_1 j_2} - 1 \leq D_{:j_1} - D_{:j_2} \leq 1 - t_{j_1 j_2} \\
& D_{:j_1} + D_{:j_2} \leq 1 + t_{j_1 j_2} \\
& Q_{:j_1}^a - Q_{:j_2}^a \leq 1 - t_{j_1 j_2}, \quad \forall a \\
& (t_{j_1 j_2} - 1)M \leq \bar{r}_{:j_1}^a - \bar{r}_{:j_2}^a \leq (1 - t_{j_1 j_2})M, \quad \forall a \\
& \sum_{j_1, j_2} t_{j_1 j_2} \geq n_b - n_z,
\end{aligned} \tag{5.6}$$

where $D_{:j}$ denotes the j^{th} column of matrix D and M can be set to $\max |r_{k_1}^a - r_{k_2}^a|$. This optimization problem can be efficiently solved to its global optimum using off-the-shelf solvers like Gurobi [25]. The additional constraints on $t_{j_1 j_2}, Q^a, \bar{r}^a$ and D adopt the big-M technique and retains the important structure of Q^a and $\bar{r}^a$ as the multiplication of a matrix and the projection matrix D: D's columns are one-hot vectors, and multiplying B^a (resp. $\hat{r}^a$) by D is essentially selecting certain columns of B^a (resp. $\hat{r}^a$) and concatenating them into Q^a (resp. $\bar{r}^a$). The binary auxiliary variables $t_{j_1 j_2}$ specify the connections between D's column selection behavior and Q^a (resp. $\bar{r}^a$)'s columns: when $t_{j_1 j_2} = 1$, $D_{:j_1} = D_{:j_2}$ guarantees $Q_{:j_1}^a = Q_{:j_2}^a$ and $\bar{r}_{:j_1}^a = \bar{r}_{:j_2}^a$ (meaning that D is selecting the same column from B^a for both j_1-th and j_2-th column of Q^a); when $t_{j_1 j_2} = 0$, D's j_1-th column is guaranteed to be different from its j_2-th column and there are no constraints on Q^a (resp. $\bar{r}^a$)'s corresponding columns. To the best of our knowledge, this is the only algorithm that computes globally optimal AIS and enables exact computation of the downstream policy to study the gap between the representation learning and the task performance.

5.2 Infinite Belief Space

We extend our discrete representation learning framework to infinite belief settings. We propose two approaches with function approximations to handle the infinitely many belief states.

5.2.1 Gradient-Based Optimization

We parametrize the discretization map $\boldsymbol{D}$ as well as the transition and reward estimation models $\{B^a\}_{a\in\mathrm{A}}$ and $\{\hat{r}^a\}_{a\in\mathrm{A}}$ as neural networks with a set of parameters θ to minimize the DAIS loss in Eq. (5.7) using end-to-end gradient-based optimization:

$$\begin{aligned}
\min_{\theta} \quad & \sum_a \sum_t |r_t^a - \hat{r}_\theta^a D_\theta(b_t)|^2 + \|B_\theta^a D_\theta(b_t) - \bar{z}_{t+1}^a\|^2 \\
\text{s.t.} \quad & [B_\theta^a]_{ij} \geq 0, \quad \forall i, j, a \text{ and } \mathbf{1}^T B_\theta^a = \mathbf{1}^T, \quad \forall a, && (5.7) \\
& \bar{z}_{t+1}^a = \sum_b P(b_{t+1} = b | b_t, a) D_\theta(b), && (5.8)
\end{aligned}$$

where $r_t^a = \mathbb{E}[R_t | b_t, a_t = a]$. Unlike the finite belief setting where there are finitely many r_k^a and $\bar{b}_k$, r_t^a and b_t can be assumed to have a continuous spectrum of values and the loss corresponding to **(AP1)** and **(AP2)** have to be minimized through sampling. In the continuous-state POMDPs with Gaussian mixed models, i.e., the setting in Section 4.2.1, the weights, means and covariances of a Gaussian mixture characterizing a belief state are flattened and concatenated into a single vector as the input to the discretization map $\boldsymbol{D}$, which outputs a one-hot vector Z_t as the discrete representation; in visual feedback control tasks, i.e., the setting in Section 4.2.2, the images are fed into the discretization map $\boldsymbol{D}$ instantiated by a convolutional neural network followed by categorical reparametrizaiton. In order to allow backpropagation through categorical variables to adjust the parameters of $\boldsymbol{D}$, $\{B^a\}$ and $\{\hat{r}^a\}$ simultaneously, we use the Gumbel-Softmax [18] as a continuous approximation to the one-hot vector.

The discretization map $\boldsymbol{D}$ essentially aggregates beliefs into clusters based on **(AP1)** and **(AP2)** loss. The one-hot vector Z_t indicates that the current belief is assigned deterministically to the cluster corresponding to Z_t's non-zero entry. The next-time-step belief b_{t+1} given the current action has the probability of being assigned to the clusters based on the categorical distribution specified by $\bar{z}_{t+1}$.

5.2.2 Clustering-Based Optimization

Jointly optimizing D, $\{B^a\}_{a\in\mathrm{A}}$ and $\{\hat{r}^a\}_{a\in\mathrm{A}}$ as in Eq. (5.7) is highly nonconvex and generally intractable (note that even the discrete case with deterministic dynamics in the previous section requires convex reparametrization and the MIP reformulation technique). Therefore, we propose to sequentially optimize D followed by $\{B^a\}$ and $\{\hat{r}^a\}$ jointly. Because the expected reward $r_t^a = \int_s r(s,a)b_t(s)ds$ is linear in the belief, aggregating the belief states with small distances to each other helps reduce the loss associated with **(AP1)**. Similarly, because the Bayesian update Eq. (3.1) is linear in belief, starting from belief states b_t close in probability metrics and executing the same action a result in b_{t+1} close to each other. If these neighboring b_t get mapped to the same DAIS z_i and similar b_{t+1} get mapped to the same DAIS z_j, $\bar{z}_{t+1}$ will become a one-hot vector $D(b_{t+1})$ and the second term in Eq. (5.9)'s objective can be made small with $B_{ji}^a = 1$. Hence, we first find a suitable discretization D via K-means clustering under the total variation-distance metric and then solve a constrained convex optimization problem to minimize the DAIS loss:

$$\begin{aligned} \min_{\{B^a\},\{\hat{r}^a\}} \quad & \sum_a \sum_t |r_t - \hat{r}^a z_t|^2 + \|B^a z_t - \bar{z}_{t+1}\|^2 \\ \text{s.t.} \quad & B_{ij}^a \geq 0, \quad \forall i,j,a \;\text{ and }\; \mathbf{1}^T B^a = \mathbf{1}^T, \quad \forall a, \end{aligned} \tag{5.9}$$

where $z_t = D(b_t)$ is the one-hot vector representing the clusters obtained by total variation K-means clustering [7] and $\bar{z}_{t+1}$ is again computed using Eq. (5.2).

To obtain more precise discretization D, one could cluster the belief states based on Wasserstein distance. The Wasserstein-style clustering resembles the iterative process of K-means clustering: the beliefs are first coarsely grouped into n_z number of clusters according to their corresponding immediate reward and each group's barycenter can

Algorithm 1 DAIS Learning and Planning

Generate data (b_t, a_t, r_t, b_{t+1}) from rollout samples of $\{a_t, o_t\}_{t\geq 1}$ using Eq. (3.1)
if gradient-based optimization **then**
    Solve Eq. (5.7) for $\{B^a\}, D, \{\hat{r}^a\}$, using gradient-based solvers
else
    Find D via Wasserstein K-means clustering
    Solve Eq. (5.9) for $\{B^a\}, \{\hat{r}^a\}$
end if
policy, V = value_iteration $(\{B^a\}, \{\hat{r}^a\})$

be calculated by solving a linear program [7]. Each belief state is then reassigned to a new cluster whose barycenter has the smallest Wasserstein distance to that belief. The barycenter calculation and assignment steps are repeated until convergence.

5.3 Planning

One main advantage of DAIS is that we can perform exact dynamic programming, i.e., value iteration, to obtain the optimal policy for such a representation. This way, we are able to "solve the approximate model exactly". Note that other planning approaches, i.e., policy iteration, Monte-Carlo tree search, may also be used for the DAIS model, but we focus on value iteration for simplicity.The overall DAIS learning and planning pipeline is summarized in Algorithm 1.

5.4 Results

In this section, we validate our discrete representation learning framework for both finite belief space task (CheeseMaze), and infinite belief space tasks: one with continuous-state space (Corridor Navigation) and the other with high-dimensional visual inputs for robotic manipulation (Object Pile Manipulation).

5.4.1 CheeseMaze

We investigate the relationship between learning loss, model performance and DAIS dimension in the CheeseMaze example adapted from [31]. We solve the DAIS optimization program (5.6) to its global optimum using Gurobi. As baselines comparison, we replace the second term in Eq. (5.6) that enforces **(AP2)** by losses corresponding to **(AP2'ab)**, **(AP2)+(AP2'a)**, **(AP2)+(AP2'b)** and **(AP2)+(AP2'ab)** respectively. Recall that the DAIS construction loss associated with **(AP1)+(AP2)** gives a concrete bound on the suboptimality of the downstream policy, we plot the DAIS construction loss associated with **(AP1)+(AP2)** for the five different optimization programs in Fig. 5-2a. We also compute the bounds α on the loss in performance and find them orders of magnitude larger (e.g., 17.6 for $n_z = 11$ and 189.9 for $n_z = 7$) than the empirical value function approximation errors. Although the DAIS fitting loss offers a relatively loose bound on approximation errors and consequently task performance (e.g., purple curve with larger DAIS fitting loss in Fig. 5-2a can have smaller value function approximation mean squared errors in Fig. 5-2c compared with the blue curve), Fig. 5-2b shows that DAIS can still recover the optimal controller computed from the true belief states (dashed line) up to compressed dimension $n_z = 9$ by solving the DAIS model exactly with value iteration. Intuitively, the higher dimensional the discrete representation is, the more capacity it has to capture useful information to model the task. Using only **(AP2)**, we observe that the DAIS loss and value function approximation error decrease monotonically as the maximum DAIS dimension grows (Fig. 5-2). Notice that using other variants of the second term in Eq. (5.6) can result in non-monotonic changes in the DAIS loss because they are not precisely minimizing the DAIS loss **(AP1)+(AP2)** but rather some redundant surrogates. The AIS loss is zero at 15 states, which is the true cardinality of the belief state, but the value function approximation error reaches zero with just 11 states.

Predicting observations is unnecessary in the regime where the DAIS can retain the optimal sufficient statistics for planning and control. Although we do not encourage observation prediction, this can still be done in this small-scale example. We then observe that in the suboptimal regime where the low-dimensional DAIS has to sacrifice useful information, predicting the output and DAIS' deterministic evolution (i.e., enforcing conditions **(AP2'ab)**) can help decrease the value function approximation error and improve the overall performance (Fig. 5-2c). In fact, minimizing the

(AP1)) and **(AP2)**) objectives is not precisely equivalent to minimizing the value function approximation error: experiments have shown that there is a nontrivial gap between the theoretical bound and the empirical evaluation. Note that the last three optimization programs have redundancy (**(AP2'ab)** imply **(AP2)**), the additional losses can change the objective landscape and thus might offer some numerical advantages for empirical implementation.

Remark (Interpretability of DAIS in CheeseMaze): Notably, one main advantage of *discrete* AIS is that the learned representation may be readily interpretable, which can be illustrated in this example. For instance, the DAIS algorithm learns to aggregate the three belief states where it is certain about its location at the bottom left cell $b_t(s) = \mathbb{1}(s = \text{bottom left cell})$, certain about its location at the bottom right cell $b_t(s) = \mathbb{1}(s = \text{bottom right cell})$ and uncertain about its location at the bottom left or right cell with probability 0.5 each $b_t(s) = 0.5 \cdot \mathbb{1}(s = \text{bottom left cell}) + 0.5 \cdot \mathbb{1}(s = \text{bottom right cell})$. The aggregation of these three beliefs does not sacrifice information for planning at all because the optimal action at all three belief states is to go north. DAIS also achieves similar aggregation for the three belief states associated with the middle left and right cells without losing any information for optimal planning (as demonstrated by the non-degrading performance and small value function approximation error until $n_z = 11$ in Fig. 5-2). One interesting observation is that if we specify DAIS' exact dimension (instead of its maximum dimension), there can be an increase in DAIS loss as the number of DAIS grows based on the problem's structure. This is because one of the three equivalent beliefs might have to be separated from the other two as the exact DAIS dimension grows (and thus breaking the symmetry of DAIS model established by the coarser aggregation). To specify exact DAIS dimension, one can add the constraint $\boldsymbol{D}\mathbf{1} \geq 1$ to Eq. (5.6), that is, each z_i gets assigned at least one $\bar{b}_k$.

Remark (Minimality of State Representation): When implementing the optimal controller, it is possible to fully describe the evolution of this particular map using only 7 controller states by aggregating the belief states with the same optimal action and analyzing their closed-loop transitions. However, this closed-loop state space is

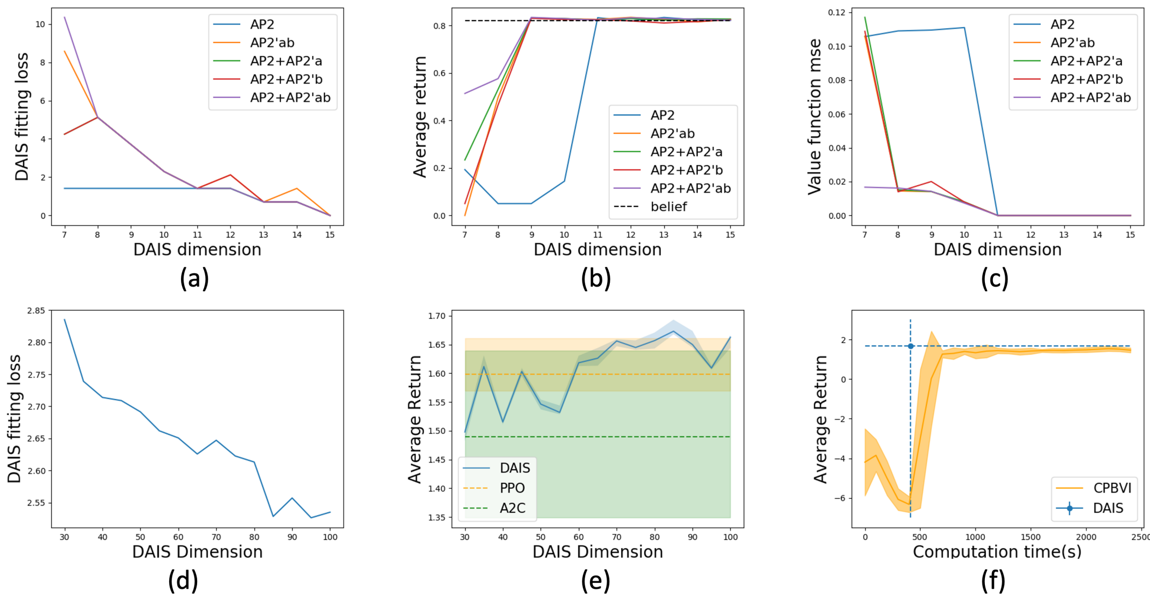


Figure 5-2: (top) DAIS loss, task performance from empirical rollouts and value function approximation error vs maximum DAIS dimension in CheeseMaze example. (bottom) DAIS loss and task performance vs exact DAIS dimension for Corridor Navigation with 10 random seeds. DAIS achieves higher return, more robust performance and much smaller runtime compared to CPBVI.

insufficient for describing transitions under policies other than the optimal one. As **(AP2)**/**(AP2'ab)** require the distribution bound to hold for *all* possible histories and actions, for the histories and actions that are not covered by the optimal policy, the error bound δ (or δ^o) can be vacuous. In other words, this 7-state representation cannot be properly characterized by the AIS framework, confirming that the AIS conditions are only sufficient but not necessary for optimal decision-making. An interesting observation is that if we decrease the weights for the transition loss corresponding to the belief states executing the same optimal action in the handcrafted controller, the optimization program with loss **(AP2)**+**(AP2'ab)** will be able to find a 8-state DAIS that recovers the optimal policy.

5.4.2 Corridor Navigation

We test the effectiveness of DAIS on the robot corridor navigation task used in [27], which fits in the setting in Sec. 4.2.1. We observe that the clustering-based optimization approach leads to better and more consistent performance than gradient-based

	DAIS + value iteration (ours)	Continuous IS + RL
model	approximate	exact
policy	exact	approximate

Table 5.1: Comparison of optimal decision making between DAIS and continuous information states.

optimization in this example, and thus report results from the former in Fig. 5-2. As an alternative to computing $\bar{z}_{t+1}$ analytically following Eq. 5.2, we can approximate $P(Z_{t+1}|Z_t, a_t)$ using samples:

$$P(Z_{t+1} = z_i | Z_t = z_j, a_t = a) \approx \frac{\sum_t \mathbb{1}(D(b_{t+1}) = z_i))\mathbb{1}(a_t = a)}{\sum_t \mathbb{1}(D(b_t) = z_j))\mathbb{1}(a_t = a)}.$$

The optimization program that uses samples to approximate the DAIS transition kernel then becomes:

$$\begin{aligned} \min_{\{B^a\},\{\hat{r}^a\}} \quad & \sum_a \sum_t |r_t - \hat{r}^a z_t|^2 + \|B^a z_t - D(b_{t+1})\|^2 \\ \text{s.t.} \quad & B^a_{ij} \geq 0, \quad \forall i, j, a \ \text{ and } \ \mathbf{1}^T B^a = \mathbf{1}^T, \quad \forall a, \end{aligned} \tag{5.10}$$

which replaces analytic $\bar{z}_{t+1}$ in program (5.9) with samples $D(b_{t+1})$.

Our method is compared with continuous point-based value iteration (CPBVI), a competitive baseline solution proposed in [27]. CPBVI samples belief points to perform Bellman updates due to the piecewise-linearity of the value function. In contrast, our method first aggregates the belief states into discrete variables and then runs value iteration on this finite set. In our experiments, we observe that CPBVI is extremely sensitive to environmental parameters, takes a long time to train and can fail to converge for certain model parameters. On the contrary, our DAIS does not suffer from convergence issues and consistently achieves higher return and much lower variance.

We also compare DAIS + value iteration against continuous information states + reinforcement learning (RL). We feed the continuous belief states (i.e. the information states with both ϵ and δ equal to 0 in **(AP1)** and **(AP2)**) into state-of-the-art RL algorithms such as PPO and A2C to learn a policy. The RL implementation is using Stable-Baselines3 [29] across 10 random seeds. As shown in Table 5.1, the *discrete* approximate information states incur approximation error for the model representation but enable quick synthesis of the optimal controller for the approximate model; the *continuous* information states achieve zero AIS construction loss but make it much harder to obtain the optimal policy due to function approximation. Fig. 5-2e shows that for tasks with certain structures (e.g. the innate discreteness in corridor navigation), DAIS can achieve better performance with much lower variance than running PPO and A2C on continuous information states.

As in any K-means approach, convergence to local optima is possible, given different initialization. Moreover, increasing $\boldsymbol{K}$ does not necessarily improve test prediction due to various factors attributable to overfitting [15]. Nevertheless, in Fig. 5-2d we observe a strong downward trend in the loss as the dimension of the DAIS is allowed to increase. As expected, this downward trend in loss correlates with an upward trend in average return. Though the increase in performance is less dramatic, it is important to note that the higher DAIS dimensions give a stronger a priori bound on the worst case suboptimality even if they achieve roughly the same expected return.

Remark (Interpretability of DAIS in Corridor Navigation): DAIS learns to capture the inherent discreteness in the corridor navigation problem: the agent only has to have a sense of its "high-level status" (i.e. whether it is in front of a door, in the corridor or at one of the two ends) but not necessarily its exact location (which is a continuous variable). This high-level information can be captured by the clustering method based on the distance between belief states. The agent then reasons with this "high-level" knowledge to solve a much simpler MDP and constantly rectifies its logical DAIS with incoming observations.

5.4.3 Visual-Feedback Control for Object Pile Manipulation

We are interested in manipulating a pile of objects (e.g., a pile of carrot pieces), whose movement is more "fluid" with interactions among themselves (carrot pieces colliding and pushing each other). We aim to extract such challenging evolution in image space into transitions in a discrete representation. We follow the setup developed by Suh et al. [32] where the robot manipulator is required to use a flat pusher to push the object pile into a target region. This can be viewed as an example of the setting in Section 4.2.2. In total, 2000 trajectories of length 20 are generated with randomly sampled actions (i.e. the pusher's starting and ending locations) in the Pymunk simulator. The greyscale images of the object pieces are downsampled to 32×32. The DAIS Z_t is then generated by feeding the images into a convolutional neural network followed by a Gumble-Softmax [18, 22]. The one-hot vector Z_t then goes through a feedforward neural network B^a with softmax as the last layer to output the categorical distribution of DAIS $\bar{z}_{t+1}$ at the next time step. The parameters of all the neural networks are optimized using the end-to-end gradient-based method proposed in Section 5.2.1. Fig. 5-3 shows that reasoning in the low-dimensional DAIS space without reconstructing or predicting the high dimensional visual outputs enables the robot manipulator to push the object pile into different target sets including circles, H-shaped and T-shaped regions. In fact, DAIS learns to aggregate images with similar block structures into the same representation catalogue.

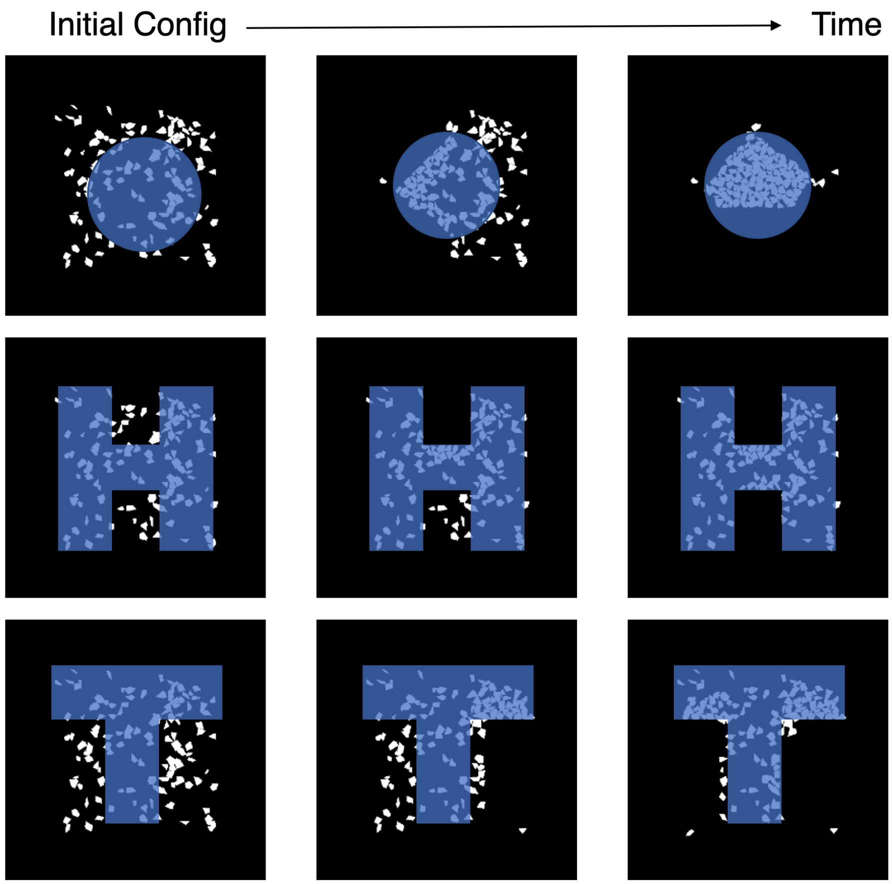


Figure 5-3: DAIS performance in object pile manipulation task: the robot manipulator is required to push the object pieces into the blue region.

Chapter 6

Conclusion

In this thesis, we evaluate discrete task-relevant representations for planning and control in partially observable environments using AIS. For finite-belief space tasks, we formulate a mixed-integer program for solving the globally optimal DAIS in terms of expected reward and Markovian transition prediction; in infinite-belief space tasks, we develop new gradient-based and clustering-based optimization methods to learn the discrete approximate representation. Even the simple finite CheeseMaze example demonstrates that the AIS bounds on closed-loop performance can be loose. However, we posit that DAIS can still be effective due to its ability to extract the most relevant information to accomplish the tasks, which often times can be characterized in a discrete form, especially for control tasks with certain structures. We are interested in validating the effectiveness of DAIS on the real robot and other partially observable robotic control tasks in the future.

Bibliography

[1] Francesco Amato, Marco Ariola, and Carlo Cosentino. Finite-time stabilization via dynamic output feedback. *Automatica*, 42(2):337–342, 2006.

[2] Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

[3] Dimitri Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.

[4] Yong-Yan Cao, James Lam, and You-Xiam Sun. Static output feedback stabilization: an ilmi approach. *Automatica*, 34(12):1641–1645, 1998.

[5] Anthony R Cassandra, Michael L Littman, and Nevin Lianwen Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. *arXiv preprint arXiv:1302.1525*, 2013.

[6] Pablo Samuel Castro, Prakash Panangaden, and Doina Precup. Equivalence relations in fully and partially observable markov decision processes. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[7] Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

[8] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004.

[9] Amalia Foka and Panos Trahanias. Real-time hierarchical POMDPs for autonomous robot navigation. *Robotics and Autonomous Systems*, 55(7):561–571, 2007.

[10] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491. PMLR, 2021.

[11] Jyotirmay Gadewadikar, Frank L Lewis, and Murad Abu-Khalaf. Necessary and sufficient conditions for h-infinity static output-feedback control. *Journal of guidance, control, and dynamics*, 29(4):915–920, 2006.

[12] Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

[13] Jacob Goldberger and Sam T Roweis. Hierarchical clustering of a mixture model. In *Advances in neural information processing systems*, pages 505–512. Citeseer, 2005.

[14] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019.

[15] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16:281–288, 2004.

[16] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Grasping POMDPs. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4685–4692. IEEE, 2007.

[17] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, pages 2117–2126. PMLR, 2018.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[20] PR Kumar and Pravin Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*, volume 75. SIAM, 1986.

[21] Alex Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33, 2020.

[22] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[23] R Andrew McCallum. Overcoming incomplete perception with utile distinction memory. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 190–196, 1993.

[24] Remi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Machine learning*, 49(2):291–323, 2002.

[25] Gurobi Optimization. Inc.,"gurobi optimizer reference manual," 2015, 2014.

[26] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032. Citeseer, 2003.

[27] Josep M Porta, Nikos Vlassis, Matthijs TJ Spaan, and Pascal Poupart. Point-based value iteration for continuous pomdps. 2006.

[28] Chunjiang Qian and Wei Lin. Output feedback control of a class of nonlinear systems: a nonseparation principle paradigm. *IEEE Transactions on Automatic Control*, 47(10):1710–1715, 2002.

[29] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. https://github.com/DLR-RM/stable-baselines3, 2019.

[30] Matthijs TJ Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research*, 24:195–220, 2005.

[31] Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *arXiv preprint arXiv:2010.08843*, 2020.

[32] HJ Suh and Russ Tedrake. The surprising effectiveness of linear models for visual foresight in object pile manipulation. *arXiv preprint arXiv:2002.09093*, 2020.

[33] Vassilis L Syrmos, Chaouki T Abdallah, Peter Dorato, and Karolos Grigoriadis. Static output feedback—a survey. *Automatica*, 33(2):125–137, 1997.

[34] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

[35] Ward Whitt. Approximations of dynamic programs, i. *Mathematics of Operations Research*, 3(3):231–243, 1978.

[36] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021.

[37] Amy Zhang, Zachary C Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, and Tommaso Furlanello. Learning causal state representations of partially observable environments. *arXiv preprint arXiv:1906.10437*, 2019.

[38] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.