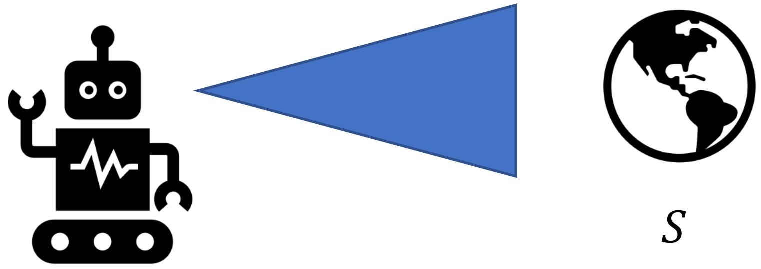
Learning Causal State Representations of Partially Observable Environments

Amy Zhang, Zachary C. Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, Tommaso Furlanello

Markov Decision Processes

- State space S
- Action space A
- Transition probability distribution P
- Reward function R

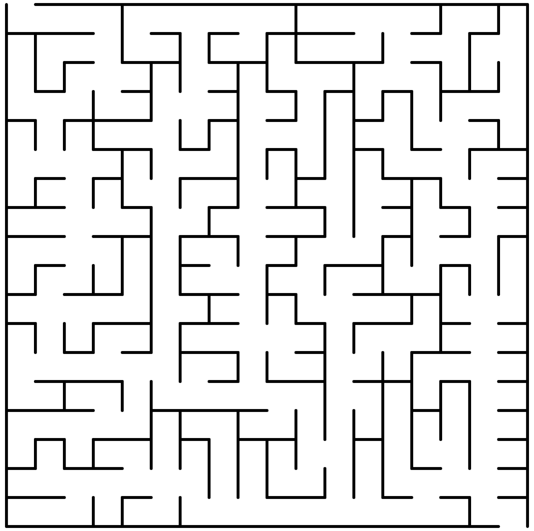


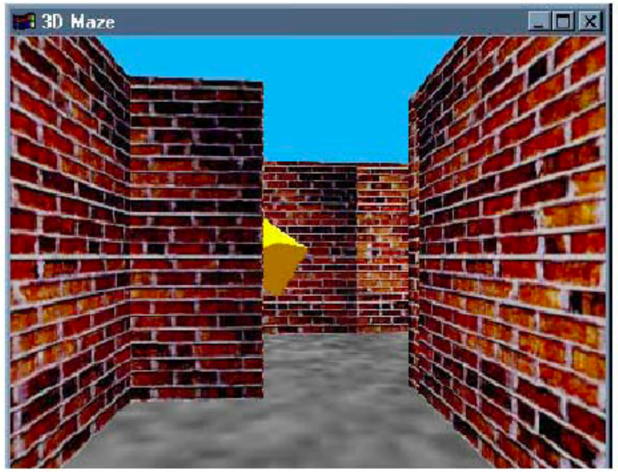
Definition: *A state has the* ***Markov Property*** *if state s_t contains all the information from the past necessary to predict the future.*

$$\Pr\{S_{t+1} = s', R_{t+1} = r | S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$$
$$= \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}$$

What if we don’t have enough information?

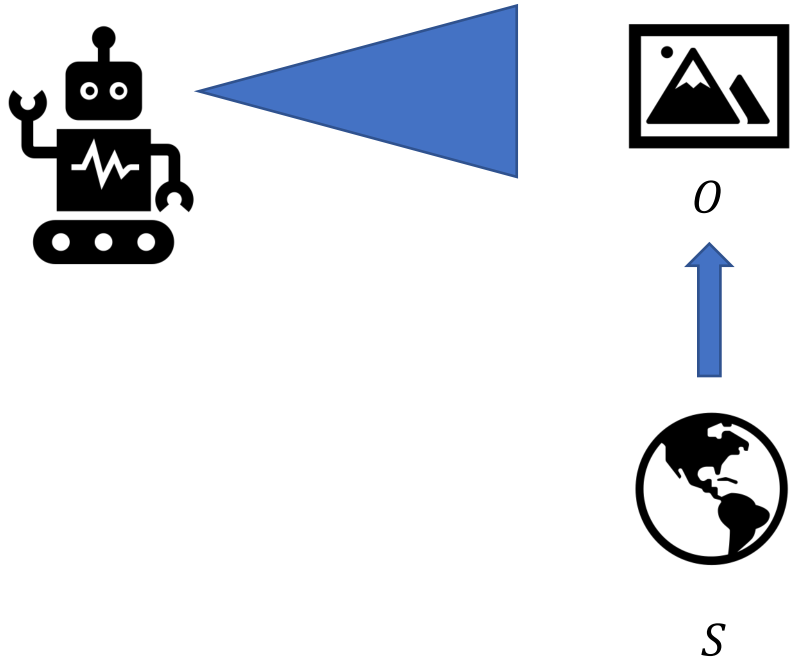
- The Markov property is a strong assumption.
- Most real world environments and problems do not give Markov observations.





Partially Observable MDPs

- State space S
- Action space A
- Transition probability distribution P
- Reward function R
- Observation space O



We no longer know what state we're in!

States are still Markovian, but observations are not.

How do we improve on observations?

Belief States

Optimal Control of Markov Processes with Incomplete State Information

K. J. ÅSTRÖM

Planning and acting in partially observable stochastic domains

Leslie Pack Kaelbling[a,*,1,2], Michael L. Littman[b,3], Anthony R. Cassandra[c,1]

Point-based value iteration: An anytime algorithm for POMDPs

Joelle Pineau, Geoff Gordon and Sebastian Thrun
Carnegie Mellon University
Robotics Institute

Predictive State Representations

Learning Predictive State Representations

Satinder Singh
Department of Electrical Engineering and Computer Science
University of Michigan
baveja@eecs.umich.edu

Michael L. Littman
Department of Computer Science
Rutgers University
mlittman@cs.rutgers.edu

Richard Sutton
Stow Research
Chester, New Jersey
rich@richsutton.com

Peter Stone
Department of Computer Sciences
The University of Texas at Austin
pstone@cs.utexas.edu

Predictive State Representations: A New Theory for Modeling Dynamical Systems

Satinder Singh **Michael R. James** **Matthew R. Rudary**

Planning with Predictive State Representations

Michael R. James
University of Michigan
mrjames@umich.edu

Satinder Singh
University of Michigan
baveja@umich.edu

Michael L. Littman
Rutgers University
mlittman@cs.rutgers.edu

Causal States

Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences

Cosma Rohilla Shalizi
Center for the Study of Complex Systems
University of Michigan
Ann Arbor, MI 48109
cshalizi@umich.edu

Kristina Lisa Shalizi
Statistics Department
University of Michigan
Ann Arbor, MI 48109
kshalizi@umich.edu

Computational Mechanics: Pattern and Prediction, Structure and Simplicity

Cosma Rohilla Shalizi* and James P. Crutchfield
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501

Computational Mechanics of Input-Output Processes: Structured transformations and the ϵ-transducer

Nix Barnett[1,2,*] and James P. Crutchfield[1,2,3,†]

Belief States

Definition: ***Belief states*** *are a posterior distribution over states.*

$$b'(s') = p(s'|a,o,b) = \frac{p(o|s',a,b)p(s'|a,b)}{p(o|a,b)}$$

$$p(o|s',a,b) = p(o|s')$$

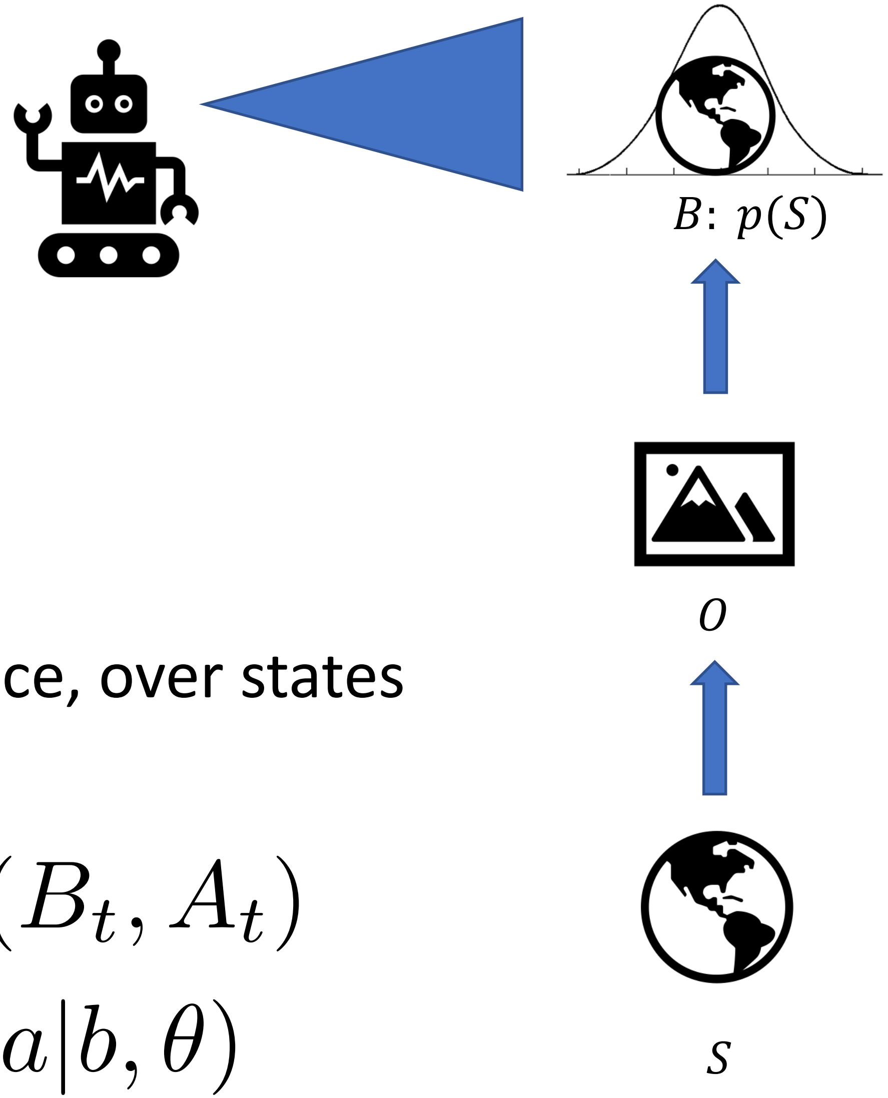
$$p(s'|a,b) = \sum_{s \in S} p(s'|a,s)b(s)$$

$$p(o|a,b) = \sum_{s' \in S} p(o|s')p(s'|a,b)$$

Assumption: The state space is known.

Belief State MDPs

- Continuous state space B
  - probability distribution over S
- Action space A
- Transition probability distribution P
- Reward function R



The Markov property holds again at convergence, over states which are distributions over the original state.

$$Q(O_t, A_t) \rightarrow Q(B_t, A_t)$$

$$\pi(a|o, \theta) \rightarrow \pi(a|b, \theta)$$

An alternative view to RL: Predictive State Representations

Predictive machines that ground representations in the history of observations

Make no assumptions about the underlying state space

Especially useful when you have issues of partial observability and state aliasing

Predictive State Representations

Definition**: (Littman, Sutton, & Singh, 2002)* ***Predictive state representations *are vectors of predictions for a specially selected set of action–observation sequences, called tests.*

A *history-based* representation, instead of depending on the ground truth states.

PSRs are a **sufficient statistic** for all future action-observation sequences.

Learning PSRs: Formulation

- System-dynamics matrix D where $D_{ij} = p(t_j | h_i)$
- probability of test $t_j = a^1 o^1 ... a^n o^n$ given a history

$$h_i = a^1 o^1 ... a^m o^m$$

independent tests = rank of D

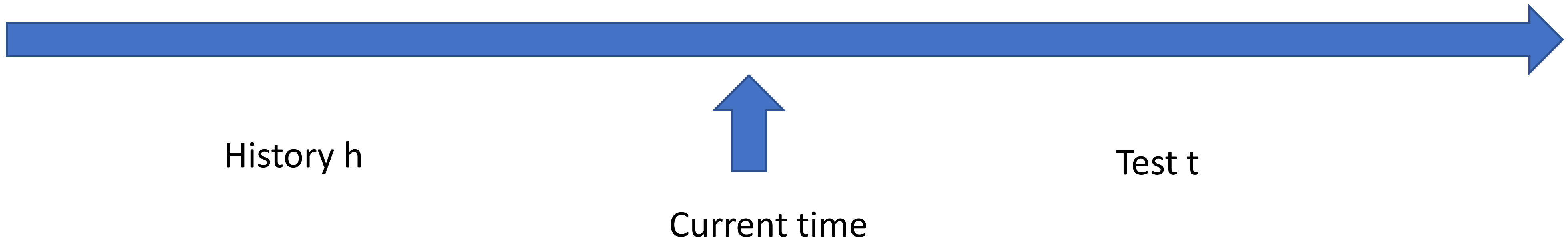
$$
\begin{array}{c|cccc}
 & t_1 & \dots & t_j & \dots \\
\hline
h_1 & p(t_1|h_1) & & p(t_j|h_1) & \\
\vdots & & & & \\
h_i & p(t_1|h_i) & & p(t_j|h_i) & \\
\vdots & & & & \\
\end{array}
$$

Learning PSRs

Core tests (linearly independent columns of D):

$$Q = \{q_1, \ldots, q_k\}$$

$p(Q|h)$ is a sufficient statistic of h for p(t|h), where tests t are possible futures given history h



=> This does not scale up well

Learning PSRs with gradient-based methods

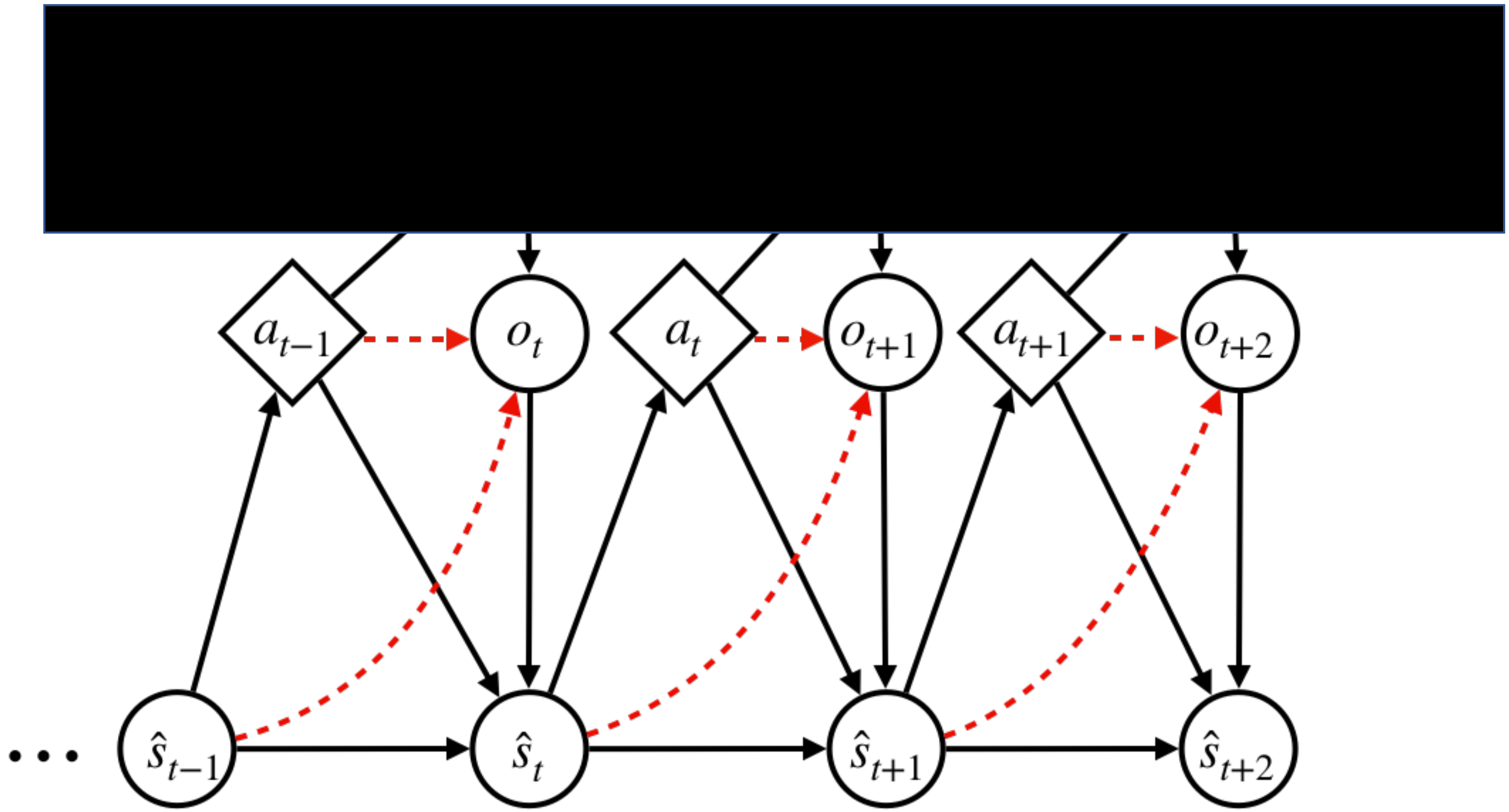
- Recurrent encoder $f : \overleftarrow{\mathbf{O}, \mathbf{A}} \mapsto \hat{\mathbf{S}}$
- Next step prediction network $\eta : \hat{\mathbf{S}} \times \mathbf{A} \mapsto \hat{\mathbf{O}}$

- We train neural network $\Psi(\overleftarrow{o, a}, a_t) = (\eta_{w_\eta} \circ f_{w_f})(\overleftarrow{o, a}, a_t)$

- Learning Objective:
  - Sufficiency:

$$\min_{w_f, w_\eta} \sum_t^T \mathcal{L}_r \big(\mathbb{P}(O_{t+1} | \overleftarrow{o, a}, a_t), \Psi(\overleftarrow{o, a}, a_t)\big)$$

Learning a Sufficient Statistic



One step further: Causality

- What is the notion of causality that is learnable in RL settings?

Definition: *A **causal model** has the ability to understand how to manipulate the world, robust to changes in behavior.*

- We want to learn causal models as opposed to a predictive model.

Expanding on PSRs: Causal States

- Stochastic process:

$\dots \quad \boxed{y_{t-1}} \quad \boxed{y_t} \quad \boxed{y_{t+1}} \quad \boxed{y_{t+2}} \quad \dots$

- Causal equivalence relation $\sim_\epsilon$

$$\overleftarrow{y} \sim_\epsilon \overleftarrow{y}' \iff \mathbb{P}(\overrightarrow{Y} \mid \overleftarrow{Y} = \overleftarrow{y}) = \mathbb{P}(\overrightarrow{Y} \mid \overleftarrow{Y} = \overleftarrow{y}').$$

- $\epsilon - map$: a mapping from past to corresponding causal state

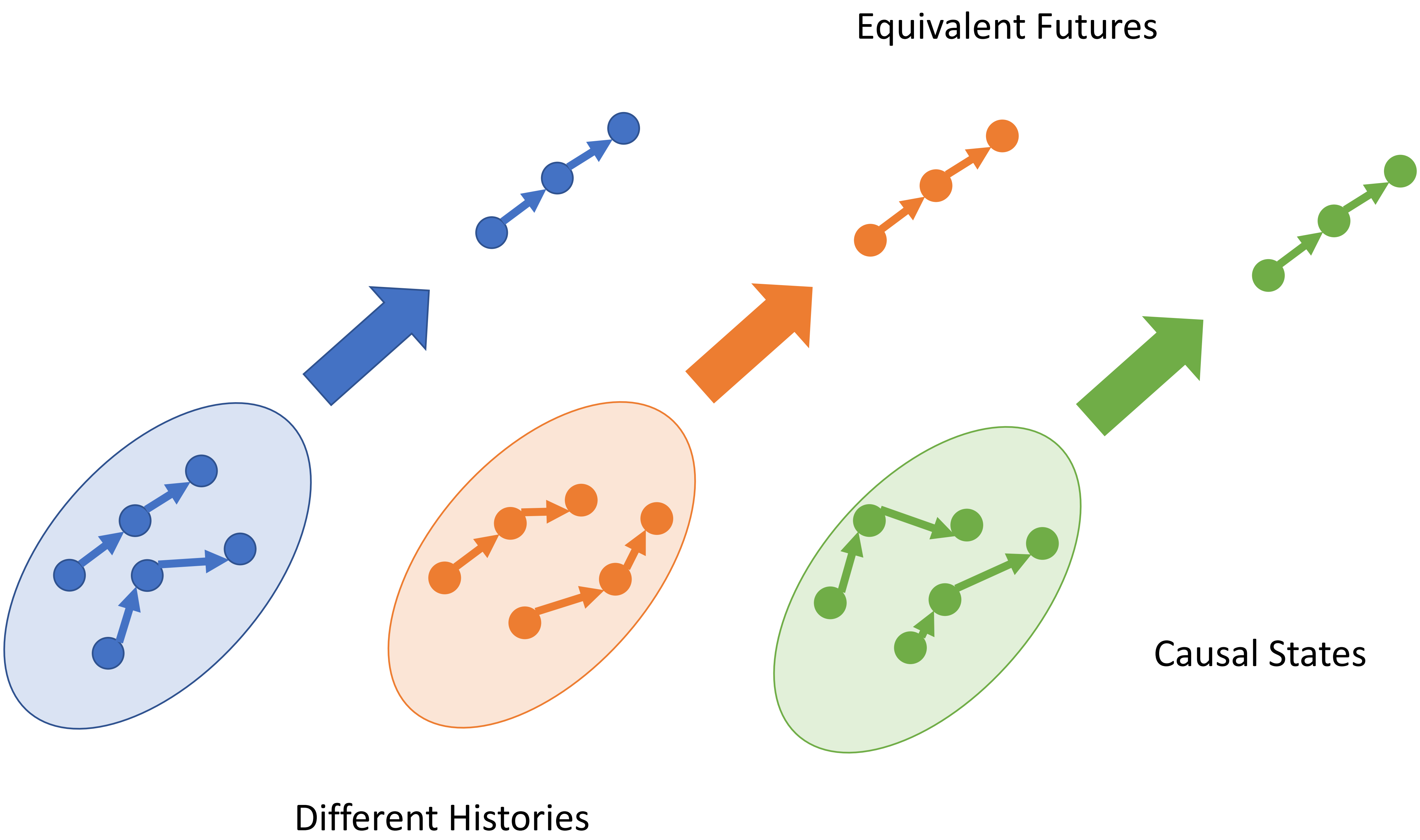
Causal State Representations

Definition 1 *(Crutchfield & Young, 1989; Shalizi & Crutchfield, 2001) The* ***causal states*** *of a stochastic process are partitions* $\sigma \in \mathbb{S}$ *of the space of feasible pasts* $\overleftarrow{\mathbf{Y}}$ *induced by the causal equivalence* $\sim_\epsilon$*:*

$$\overleftarrow{y} \sim_\epsilon \overleftarrow{y}' \iff \mathbb{P}(\overrightarrow{Y}|\overleftarrow{Y} = \overleftarrow{y}) = \mathbb{P}(\overrightarrow{Y}|\overleftarrow{Y} = \overleftarrow{y}'). \tag{1}$$

Which implies:

$$\mathbb{P}(\overrightarrow{Y}|S_t = \sigma_i) = \mathbb{P}(\overrightarrow{Y}|\overleftarrow{Y} = \overleftarrow{y}) \quad \forall \quad \overleftarrow{y} \in \sigma_i, \tag{2}$$


Equivalent Futures
Causal States
Different Histories

Our Goal

Given a stochastic process we can generate *causal states*

- Minimally sufficient in all future prediction
- Discrete states with deterministic transitions

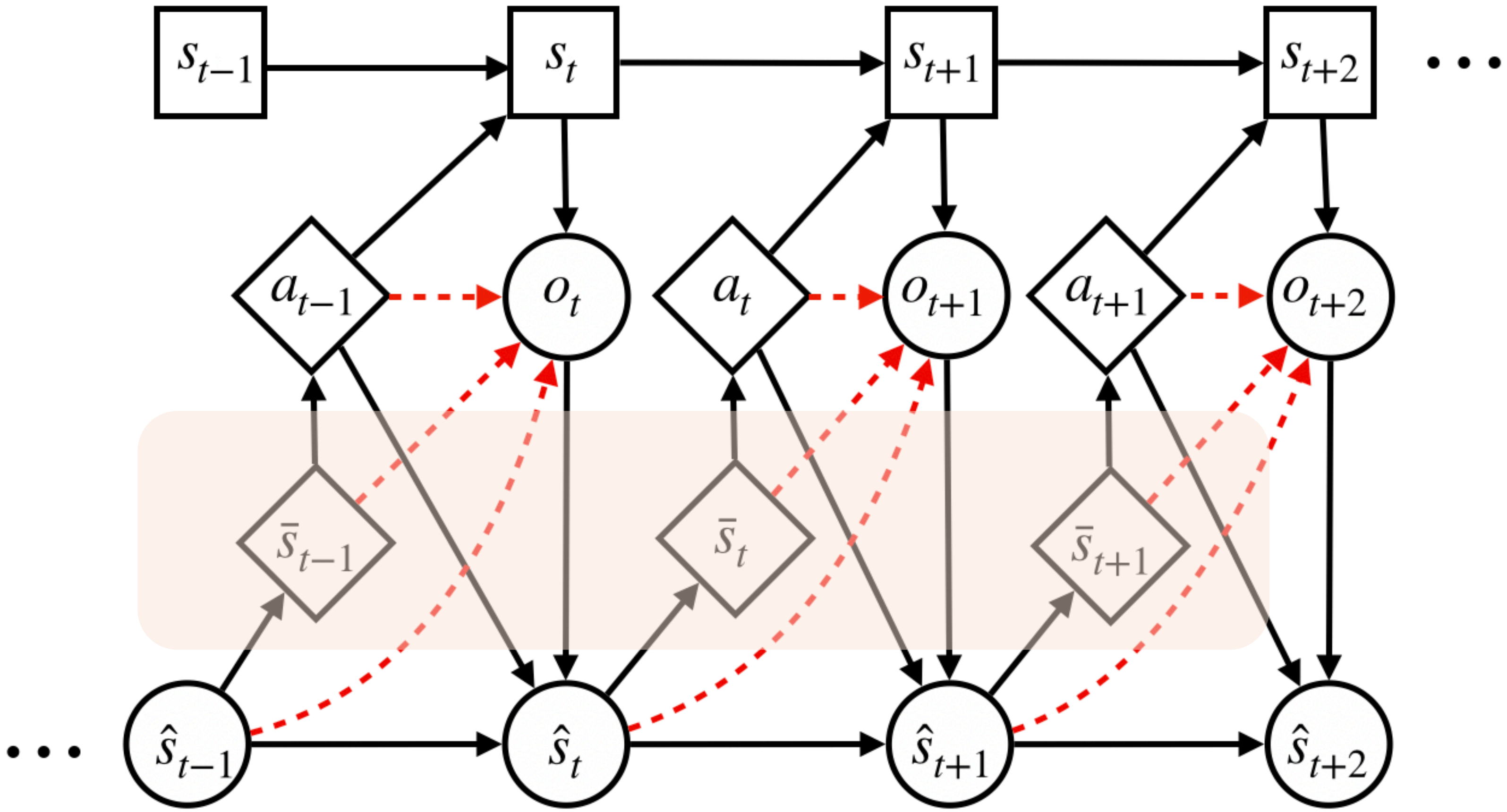
$$\mathrm{H}[S_{t+1}|Y_t, S_t] = 0.$$

- Near-Markovian

$$\begin{tikzcd}
\cdots \arrow[r, "y_{t-2}"] & \overleftarrow{y}_{t-1} \arrow[r, "y_{t-1}"] \arrow[d, "\epsilon"'] & \overleftarrow{y}_{t} \arrow[r, "y_{t}"] \arrow[d, "\epsilon"'] & \overleftarrow{y}_{t+1} \arrow[r, "y_{t+1}"] \arrow[d, "\epsilon"'] & \cdots \\
\cdots \arrow[r, "y_{t-2}"] & s_{t-1} \arrow[r, "y_{t-1}"] & s_{t} \arrow[r, "y_{t}"] & s_{t+1} \arrow[r, "y_{t+1}"] & \cdots
\end{tikzcd}$$

Method

- Minimal sufficient statistics can be computed from any other non-minimal sufficient statistic.



Components

- Recurrent encoder $f : \overleftarrow{\mathbf{O}, \mathbf{A}} \mapsto \hat{\mathbf{S}}$
- Next step prediction network $\eta : \hat{\mathbf{S}} \times \mathbf{A} \mapsto \hat{\mathbf{O}}$
- Discretizer $\bar{d}^s : \hat{\mathbf{S}} \mapsto \bar{\mathbf{S}}$
- Second prediction network – ensure sufficiency of the discretized representation $\bar{\eta} : \bar{\mathbf{S}} \times A \mapsto \mathbf{O}$

Model Architecture

- We train neural network $\Psi(\overleftarrow{o,a}, a_t) = (\eta_{w_\eta} \circ f_{w_f})(\overleftarrow{o,a}, a_t)$
- Discretizer and 2nd prediction network

$$\Lambda(\overleftarrow{o,a}, a_t) = (\bar{\eta}_{w_{\bar{\eta}}} \circ \bar{d}^s_{w_{\bar{d}}} \circ f_{w_f^*})(\overleftarrow{o,a}, a_t)$$

Learning Objectives

- Sufficiency: $\min_{w_f, w_\eta} \sum_t^T \mathcal{L}_r\big(\mathbb{P}(O_{t+1} | \overleftarrow{o,a}, a_t), \Psi(\overleftarrow{o,a}, a_t)\big)$
- Knowledge distillation: $\min_{w_{\bar{\eta}}, w_{\bar{d}}} \sum_t^T \mathcal{L}_d\big(\Psi(\overleftarrow{o,a}, a_t), \Lambda(\overleftarrow{o,a}, a_t)\big).$

Evaluation

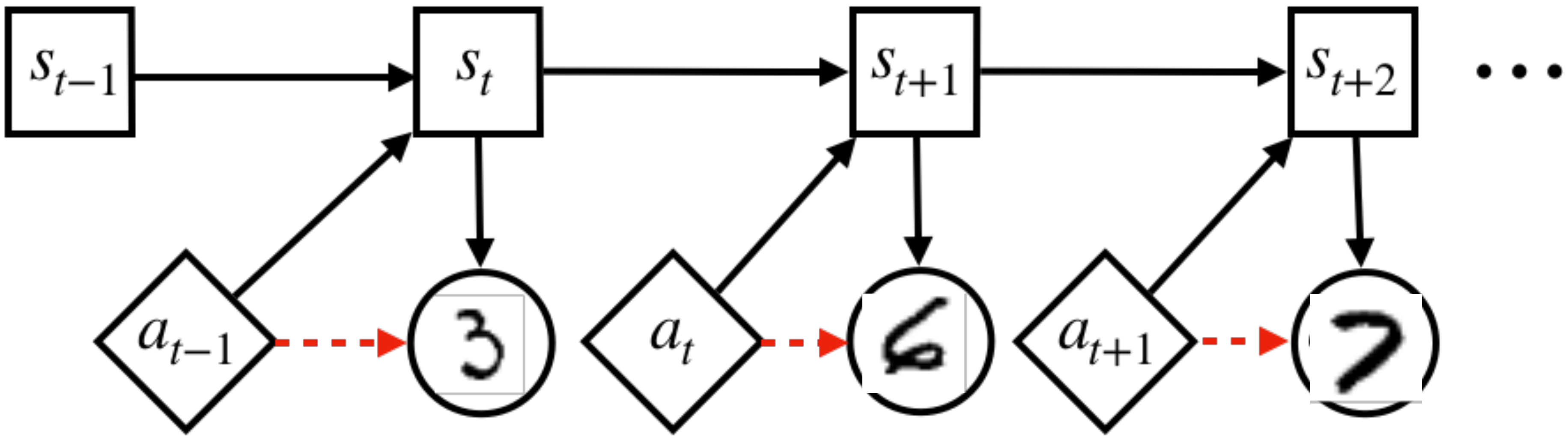
- Our learning objective is next-step prediction
- How do we show usefulness of this representation?
- We evaluate by learning downstream policies with Q-learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$

Environments

1. Stochastic processes:
    1. Discrete observation
    2. Continuous observation – stochastic rendering
    3. High dimensional observation – stochastic rendering
2. GridWorlds
3. Doom
4. Atari

Stochastic Dynamics and High-dimensional Observations



- Transition function:

$$p = 0.75$$

$$\mathbb{P}(O_{t+1} = o' | O_{t-k} = o') = p$$

$$\mathbb{P}(O_{t+1} = o' | O_{t-k} \neq o') = 1 - \frac{p}{|O|} \qquad o' \in \mathbf{O}$$

- Action space:

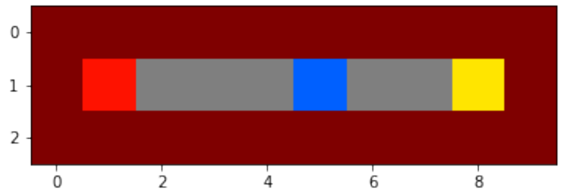
$$p(O_{t+1}=i|A_t=0)=\begin{cases} p & \text{if } o_{t-k}=i, \\ \frac{1-p}{|O|} & \text{otherwise.} \end{cases},$$

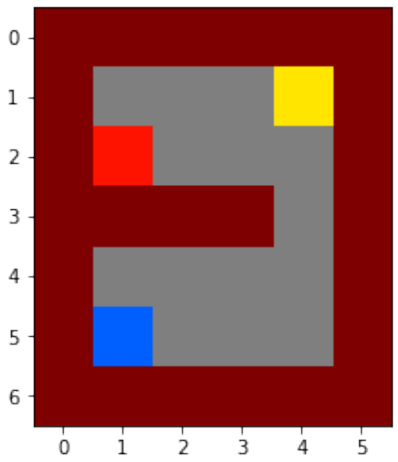
$$p(O_{t+1}=i|A_t=1)=\begin{cases} p & \text{if } o_{t-k-1}=i, \\ \frac{1-p}{|O|} & \text{otherwise.} \end{cases}.$$

- +1 reward for state = 0

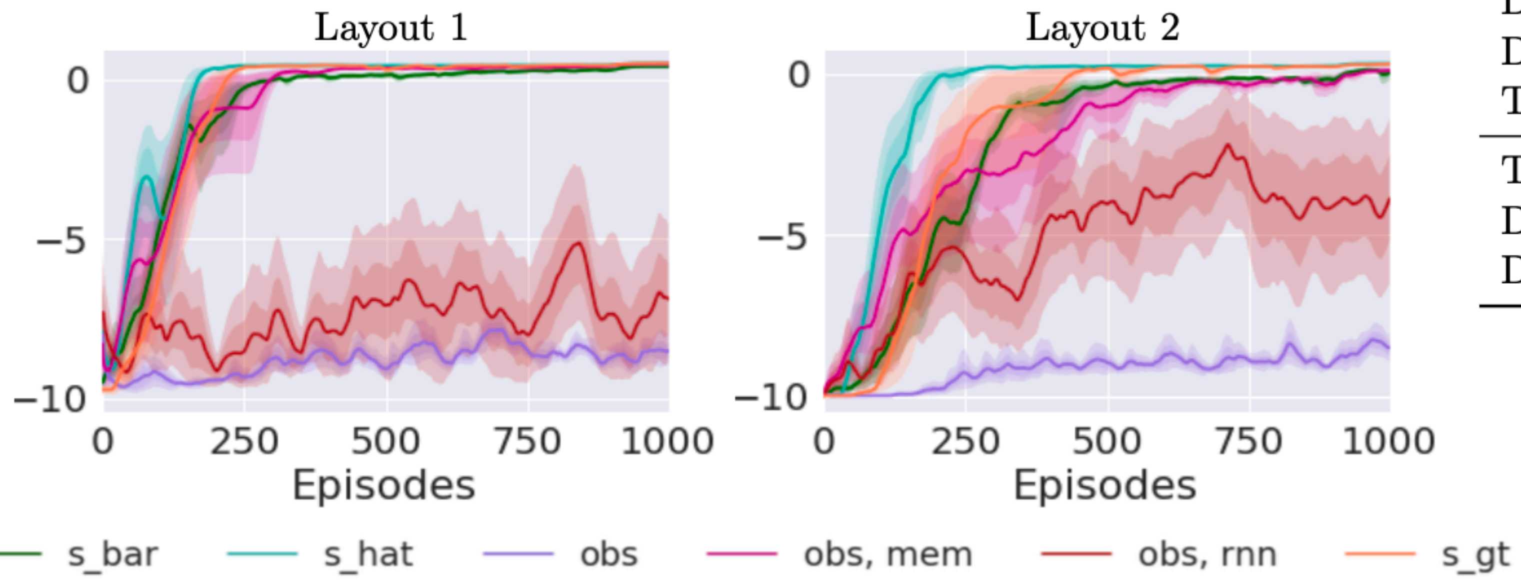
Method	Discrete $\lvert Y\rvert, k=2$	Discrete $\lvert Y\rvert, k=4$	Gaussian $\lvert Y\rvert, k=2$	Gaussian $\lvert Y\rvert, k=4$	MNIST $\lvert Y\rvert, k=2$	MNIST $\lvert Y\rvert, k=4$
DQN on Y	50.1, 1.01	25.1, 1.12	50.6, 1.26	25.0, 1.35	50.1, 1.80	25.0, 1.27
DQN on $\overleftarrow{Y}$	**73.7, 0.73**	**55.5, 1.62**	**73.3, 1.20**	**54.9, 1.71**	**72.3, 1.33**	**54.2, 1.39**
DQN on $\hat{S}$	72.7, 1.04	**54.6, 1.61**	**73.6, 0.82**	**55.3, 1.91**	**72.8, 1.23**	50.8, 1.80
DQN on $\bar{S}$	72.6, 4.10	49.2, 3.29	**73.7, 2.18**	52.7, 3.07	**72.6, 2.50**	43.2, 3.02

GridWorlds

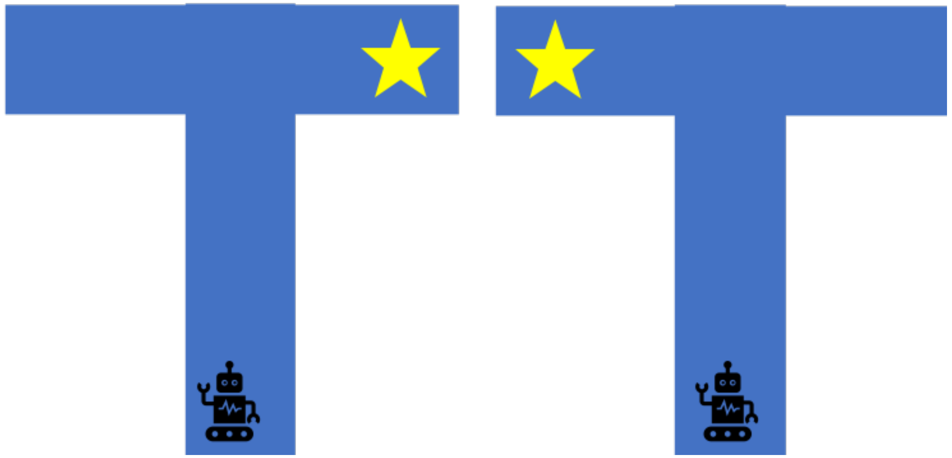


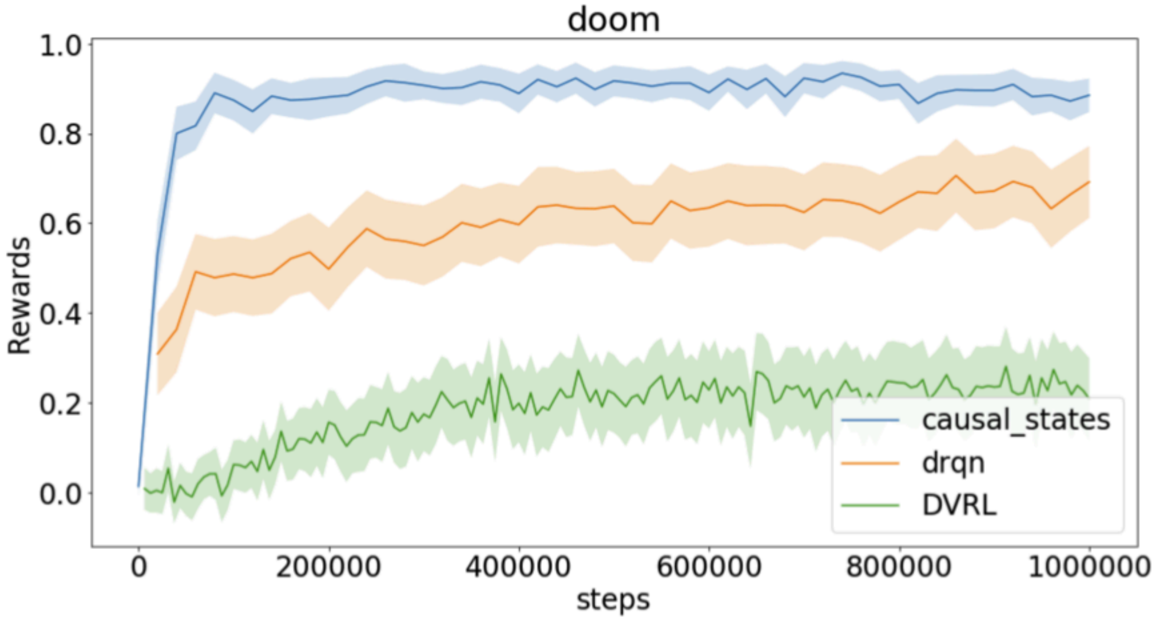


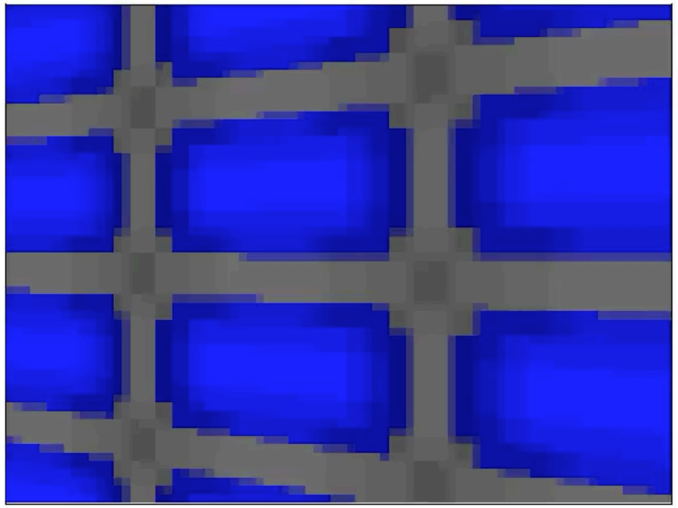
Method	Layout 1	Layout 2
Tabular, $\bar{S}$	$0.43 \pm 0.$	$0.01 \pm 0.$
DQN, $\bar{S}$	$\mathbf{0.50 \pm 0.005}$	-0.17 ± 0.24
DQN, $\hat{S}$	$\mathbf{0.5 \pm 0.}$	$\mathbf{0.30 \pm 0.}$
Dijkstra, $\bar{S}$	$\mathbf{0.5,\ 0.}$	$\mathbf{0.3,\ 0.}$
DQN, Y	-9.46 ± 0.06	-9.48 ± 0.04
DQN, $Y_{\leq t}$	-0.91 ± 0.95	0.23 ± 0.05
DRQN, Y	-9.75 ± 0.07	-5.63 ± 1.18
Tabular, Y	$-9.40 \pm 0.$	$-9.11 \pm 0.$
Tabular, S_{gt}	$0.45 \pm 0.$	$0.23 \pm 0.$
DQN, S_{gt}	0.44 ± 0.01	$\mathbf{0.30 \pm 0.003}$
Dijkstra, S_{gt}	$\mathbf{0.5,\ 0.}$	$\mathbf{0.3,\ 0.}$

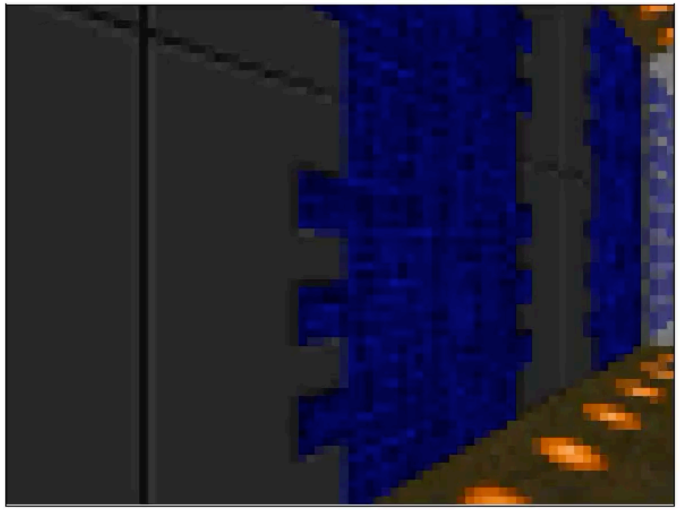


Doom Environment

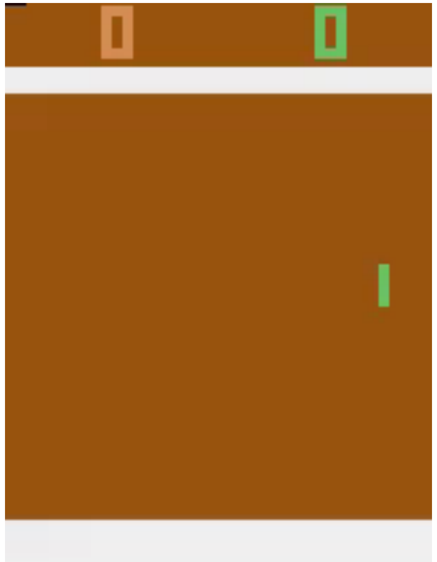


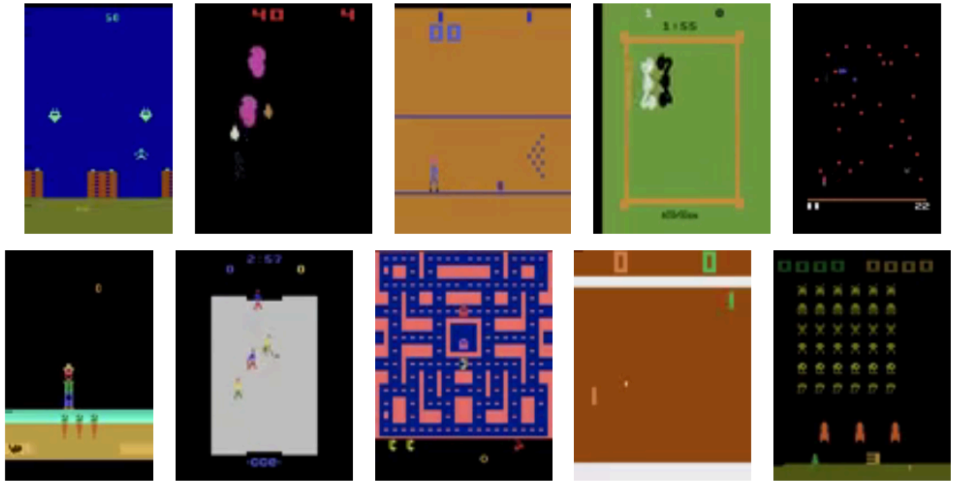






Atari

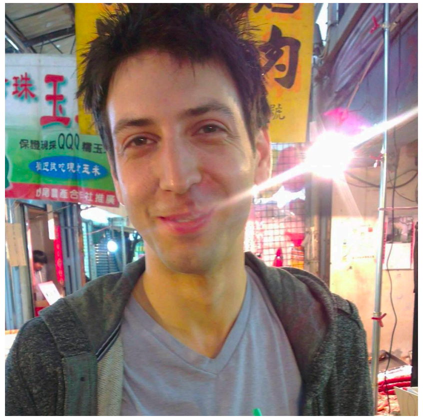




Game	Causal States	DRQN	DVRL
Air Raid	**950 ± 271**	518 ± 231	**748 ± 156**
Asteroids	**1129 ± 345**	**929 ± 285**	349 ± 54
Bowling	**34 ± 8**	**29 ± 0**	23 ± 1
Boxing	4 ± 4	0 ± 2	**16 ± 3**
Centipede	**4586 ± 763**	3127 ± 71	1157 ± 130
Gopher	**783 ± 151**	620 ± 129	255 ± 129
Ice Hockey	**−3 ± 1**	**−5 ± 1**	−11 ± 0
Ms. Pacman	671 ± 36	**849 ± 60**	181 ± 45
Pong	**−2 ± 6**	**−7 ± 7**	−20 ± 0
Space Invaders	**354 ± 67**	**381 ± 14**	68 ± 9

Contributions and Discussion

- Two contributions:
  - A gradient-based learning method for PSRs
  - A notion of causality and discretization to achieve causal states
- Discrete vs. Continuous
  - Causal states give additional interpretability
  - There’s an inherent trade-off of interpretability and performance



Arxiv: 1906.10437

Invariant Causal Prediction for Rich Observation MDPs

Amy Zhang[*,1,2,3] Clare Lyle[*,4] Shagun Sodhani[3] Angelos Filos[4] Marta Kwiatkowska[4] Joelle Pineau[1,2,3]
Yarin Gal[4] Doina Precup[1,2,5]

1 McGill

2 Mila

3

4

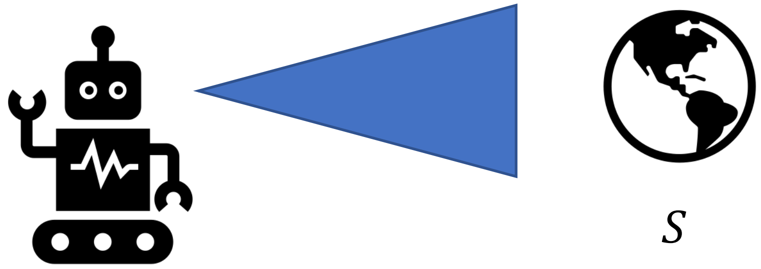
5

* Equal contribution

ICML 2020
Arxiv: 2003.06016

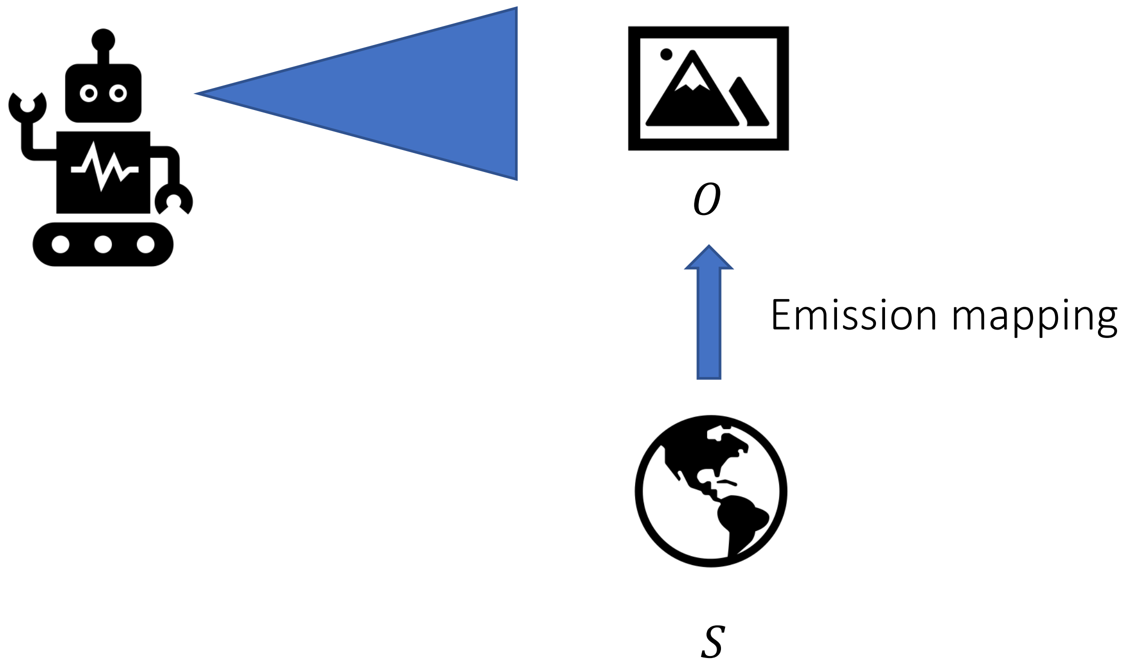
Markov Decision Processes

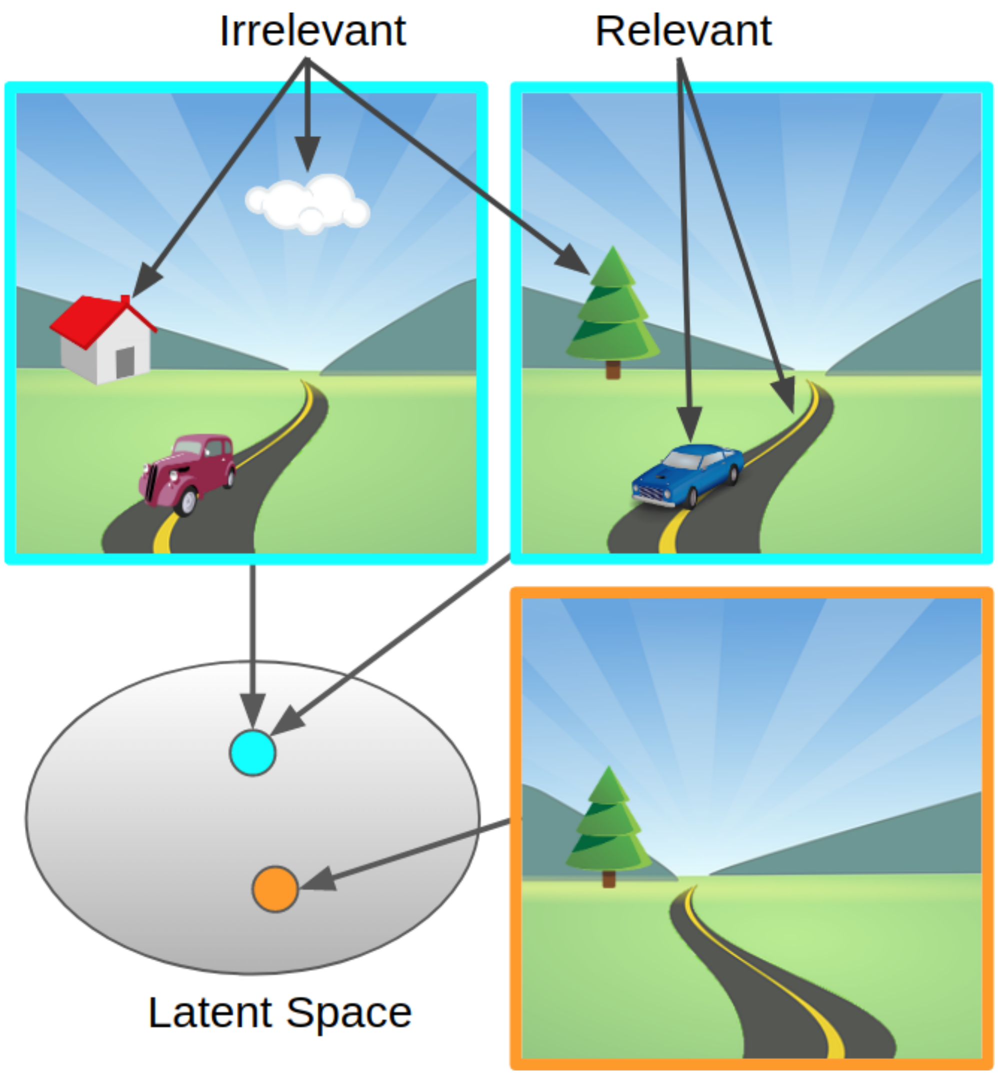
- State space S
- Action space A
- Transition probability distribution P
- Reward function R



What kind of additional structure is reasonable to assume in MDPs ?

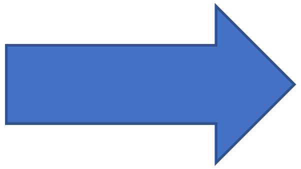
A realistic additional assumption





Motivation

- Goal: Generalization to new observations *where the underlying MDP is the same*
- Solution: Ignore irrelevant information



Motivation

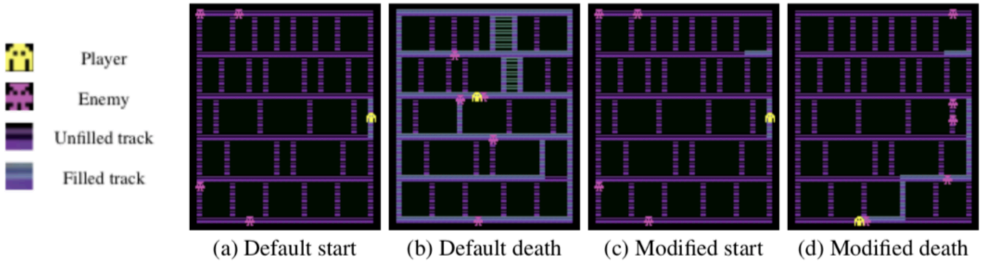


Figure: Train and Test on Atari proposed by Witty et al. 2018

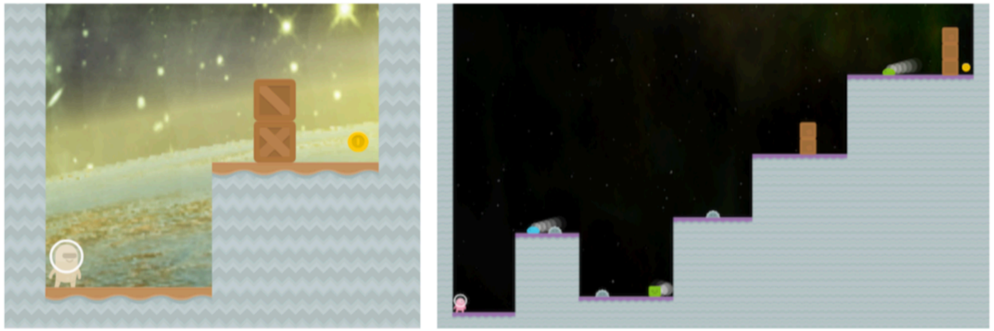


Figure: Train and Test on CoinRun proposed by Cobbe et al. 2019

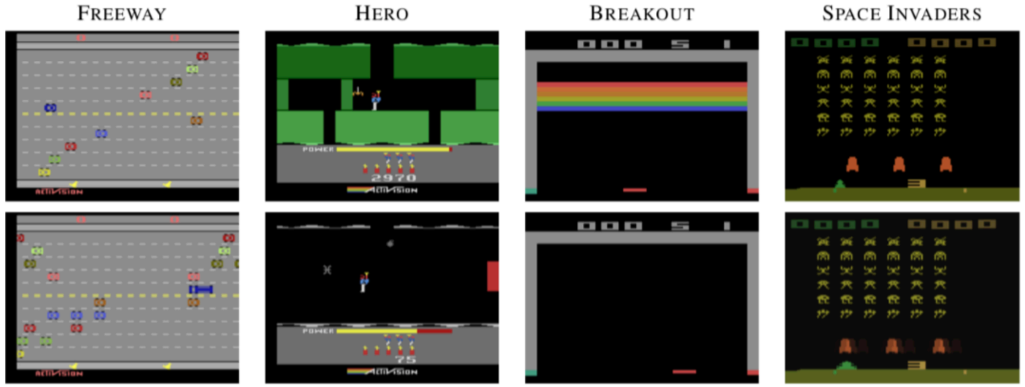
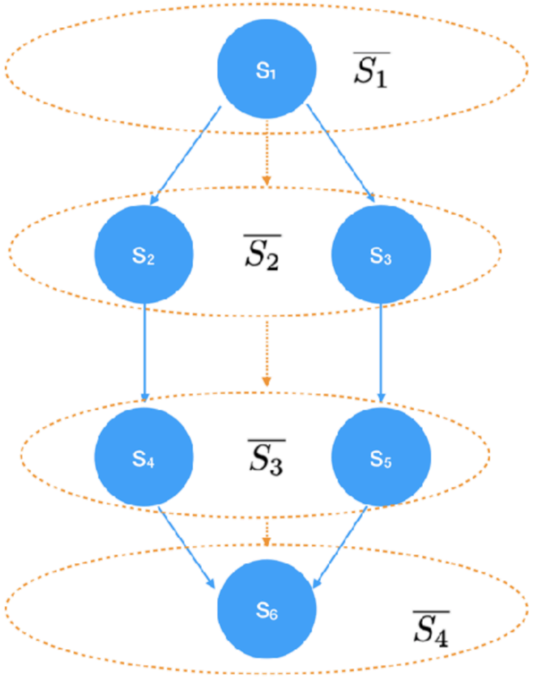


Figure: Train and Test on Atari proposed by Farebrother, Machado, and Bowling 2018[2] .

State Abstractions

A state abstraction is a function $\phi : S \mapsto \bar{S}$ which maps states $s \in S$ to simpler abstract state space $\bar{S}$. This can make it easier for an agent to learn and plan.

A *model-irrelevance state abstraction (MISA)* is a state abstraction that preserves the reward function and transition dynamics of the MDP. i.e.



$$\phi(s_1) = \phi(s_2) \implies$$

$$R(s_1) = R(s_2)$$

and

$$\sum_{s' \in \phi^{-1}(\bar{s}')} p(s'|s_1) = \sum_{s' \in \phi^{-1}(\bar{s}')} p(s'|s_2)$$

Causal Graphs (Structural Causal Models)

- Target variable: Y
- Causal feature set: X_2, X_4
- Directed arrows = causal relationship
- X_2 *causes* Y

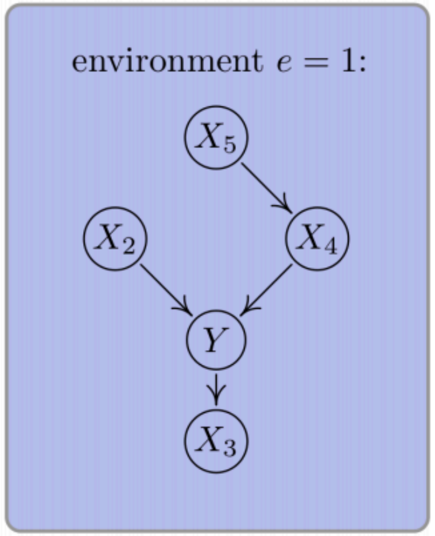


Figure from Peters et al. (2016)

Causal Inference Using Invariant Prediction

Peters et al. (2016) first introduced an algorithm, Invariant Causal Prediction (ICP), to find the causal feature set.

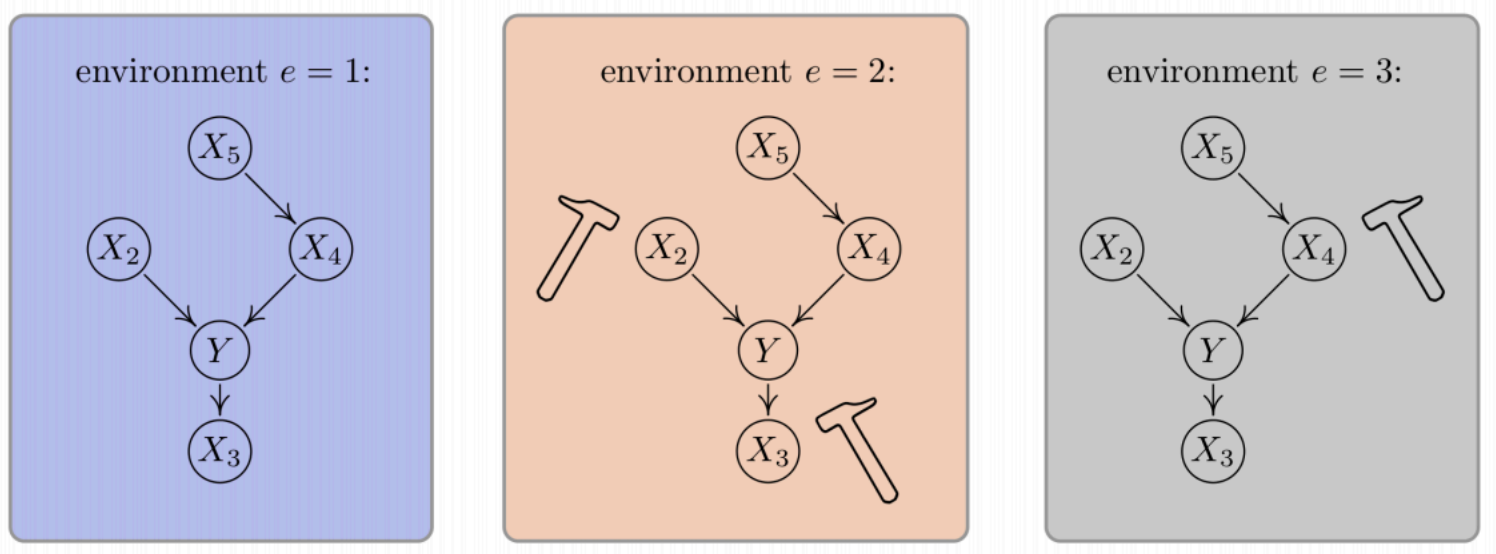


Figure 1: An example including three environments. The invariance (1) and (2) holds if we consider $S^* = \{X_2, X_4\}$. Considering indirect causes instead of direct ones (e.g. $\{X_2, X_5\}$) or an incomplete set of direct causes (e.g. $\{X_4\}$) may not be sufficient to guarantee invariant prediction.

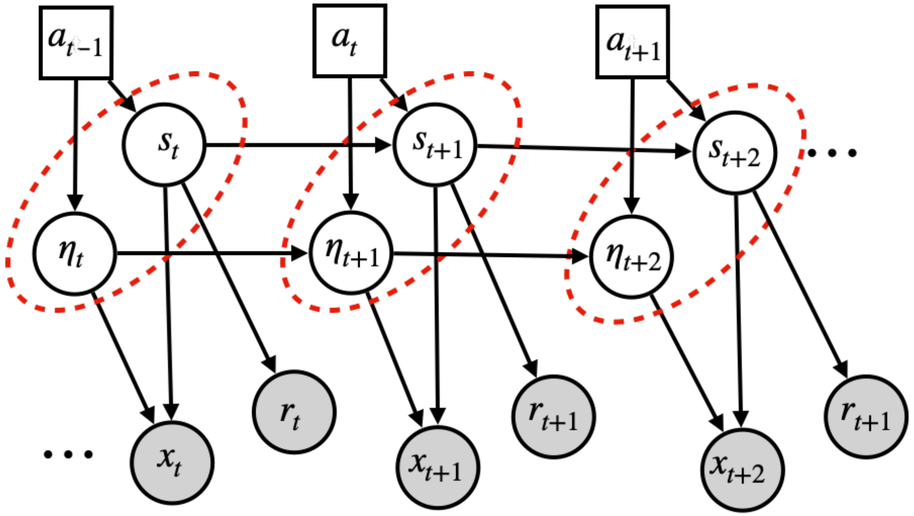
Figure from Peters et al. (2016)

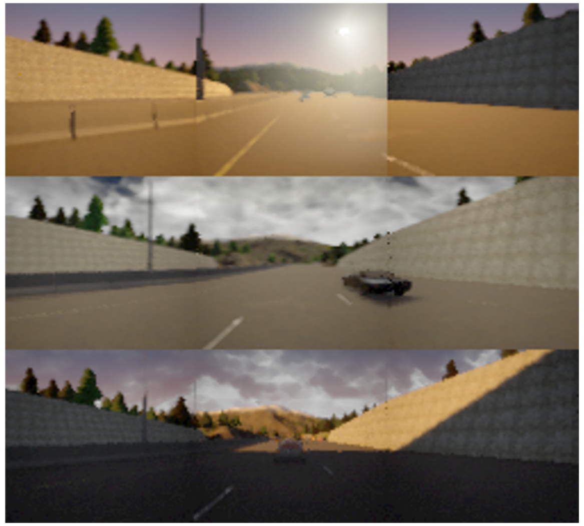
Block MDPs (Rich Observation MDPs)

Definition

A Block MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{X}, p, q, R \rangle$

- unobservable state space $\mathcal{S}$
- finite action space $\mathcal{A}$
- observation space $\mathcal{X}$
- transition distribution p
- reward function R
- emission $q : \mathcal{S} \to \mathcal{X}$



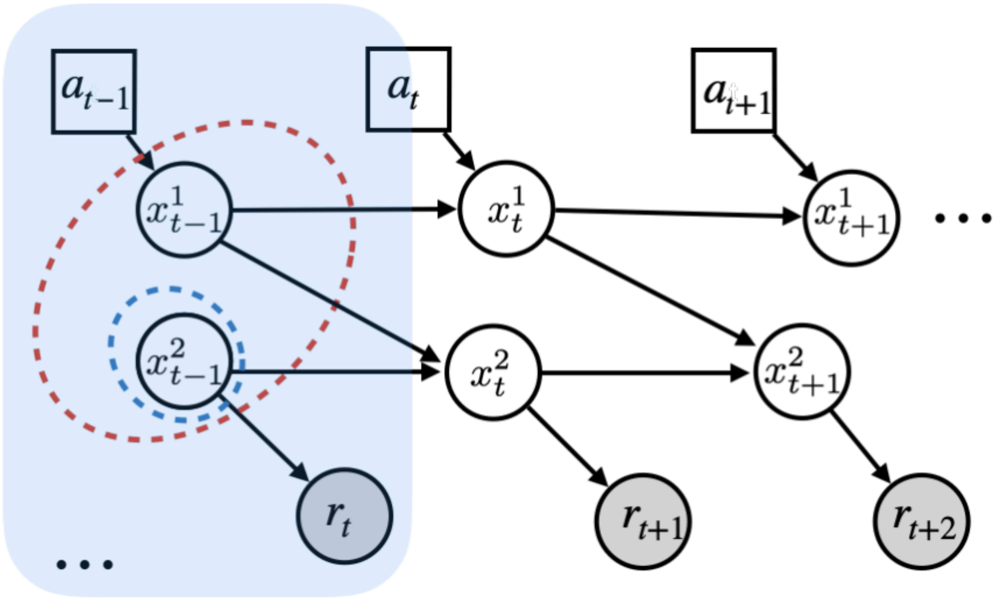


Assumptions

- **Assumption 1**: The observation space of a Block MDP is fully observable, and therefore exhibits the Markov property.
- **Assumption 2**: The components of the current observation are independent conditioned on the previous observation, i.e.

$$p(X^1_{t+1}|X_t, X^2_{t+1}) = P(X^1_{t+1}|X_t) \qquad (1)$$

- **Assumption 3**: The training environments correspond to interventions on spurious variables in the observation space.



Graphical model demonstrating Assumption 2.

Causality and MISAs

Causal Variables $\iff$ **State Abstractions**

- Consider the setting where variables are observable: state $s = (x_1, \ldots, x_n)$.
- Take the variables which are **causal ancestors** of the return, $\bar{s} = (x_{i_1}, \ldots, x_{i_k})$
- Then the mapping $\phi : (x_1, \ldots x_n) \mapsto (x_{i_1}, \ldots, x_{i_k})$ ...
  is a **model irrelevance state abstraction**

Theorem 1

Let $S_R \subseteq \{1, \ldots, k\}$ be the set of variables such that the reward $R(x, a)$ is a function only of $[x]_{S_R}$ (x restricted to the indices in S_R). Then let $S = \mathbf{AN}(R)$ denote the ancestors of S_R in the (fully observable) causal graph corresponding to the transition dynamics of $M_{\mathcal{E}}$. Then the state abstraction $\phi_S(x) = [x]_S$ is a *model-irrelevance* abstraction for every $e \in \mathcal{E}$.

Bounds on Generalization Error

Good state abstractions

MISAs generalize well to new environments because the agent can immediately apply its knowledge from previous environments.

Model error bound

Consider an MDP M, with M' denoting a coarser bisimulation of M. Let ϕ denote the mapping from states of M to states of M'. Suppose that the dynamics of M are L-Lipschitz w.r.t. $\phi(X)$ and that T is some approximate transition model satisfying $\max_s \mathbb{E}\|T(\phi(s)) - \phi(T_M(s))\| < \delta$, for some $\delta > 0$. Let $W_1(\pi_1, \pi_2)$ denote the 1-Wasserstein distance. Then

$$\mathbb{E}_{x\sim M'}[\|T(\phi(x)) - \phi(T_{M'}(x))\|] \leq \delta + 2LW_1(\pi_{\phi(M)}, \pi_{\phi(M')}). \tag{2}$$

Bounds on Generalization Error

$$
\begin{aligned}
J_R^\infty &:= \sup_{x\in\mathcal{X},a\in\mathcal{A}} |R(\phi(x),a,\phi(x')) - r(x,a)| \\
J_D^\infty &:= \sup_{x\in\mathcal{X},a\in\mathcal{A}} W_1(f_s(\phi(x),a),\phi P(x,a)).
\end{aligned} \tag{4}
$$

Theorem 3. *Let M be a block MDP and $\bar{M}$ the learned invariant MDP with a mapping $\phi : \mathcal{X} \mapsto \mathcal{Z}$. For any L-Lipschitz valued policy π the value difference of that policy is bounded by*

$$
|Q^\pi(x,a) - \bar{Q}^\pi(\phi(x),a)| \leq \frac{J_R^\infty + \gamma L J_D^\infty}{1-\gamma}, \tag{5}
$$

where Q^π is the value function for π in M and $\bar{Q}^\pi$ is the value function for π in $\bar{M}$.

Learning an abstraction

1. We first introduce a linear algorithm for learning *Model-Irrelevance State Abstractions* (MISA) – based on Peters et al. (2016).
2. We extend to nonlinear settings with a gradient-based method for disentangling the state space into a minimal representation that *causes* reward, and everything else.

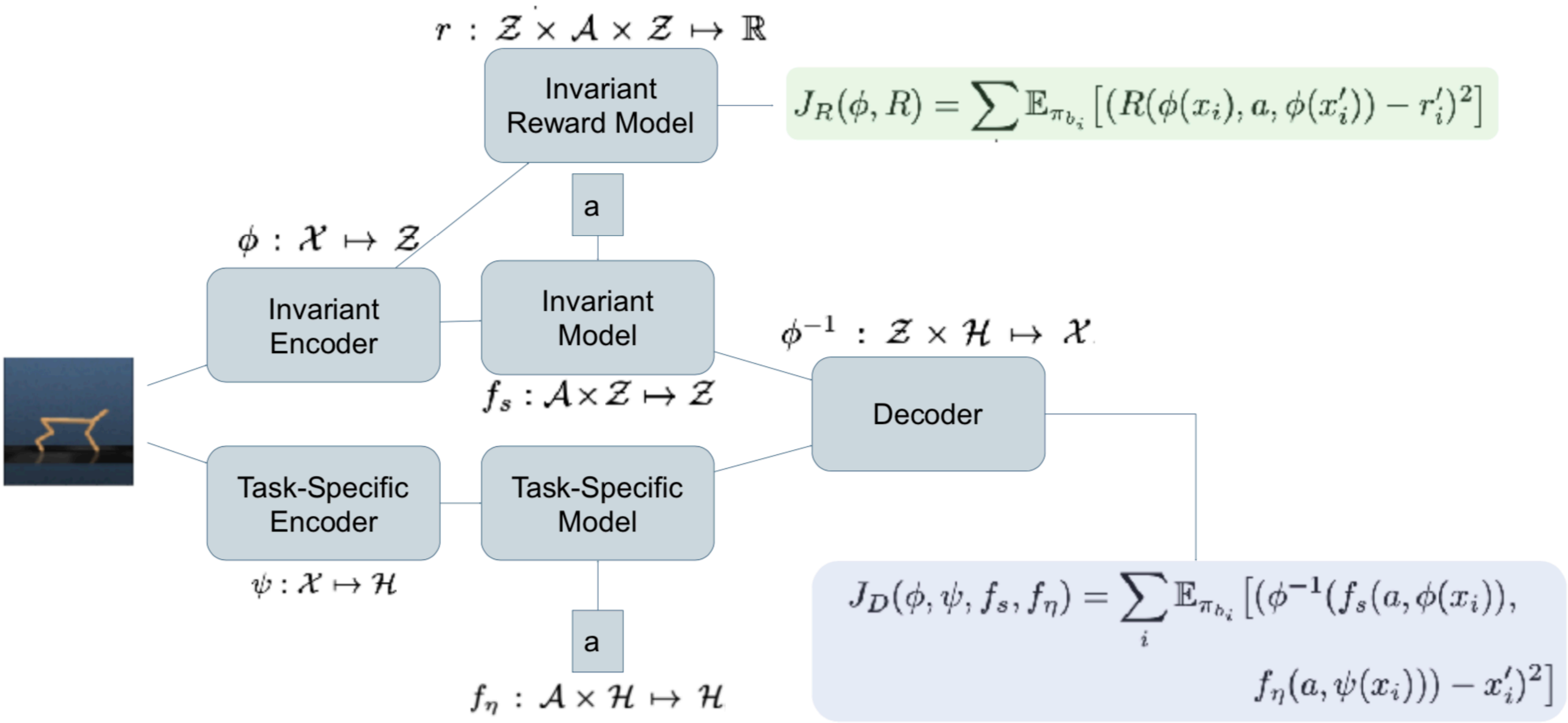
Observable Variables Setting

Algorithm: ICP for Model Irrelevance State Abstractions

Result: $S \subset \{1, \ldots, k\}$, the causal state variables
Input: α, a confidence parameter, $\mathcal{D}$, an replay buffer with observations $\mathcal{X}$ (partitioned into environments $e_1, \ldots, e_k$). $S \leftarrow \emptyset$;
stack $\leftarrow$ r ;
while *stack is not empty* **do**
  v = stack.pop() ;
  if $v \notin S$ **then**
    $S' \leftarrow \texttt{ICP}(\mathrm{v}, \mathcal{D}, \frac{\alpha}{\dim(\mathcal{X})})$;
    $S \leftarrow S \cup S'$;
    stack.push(S')
return S

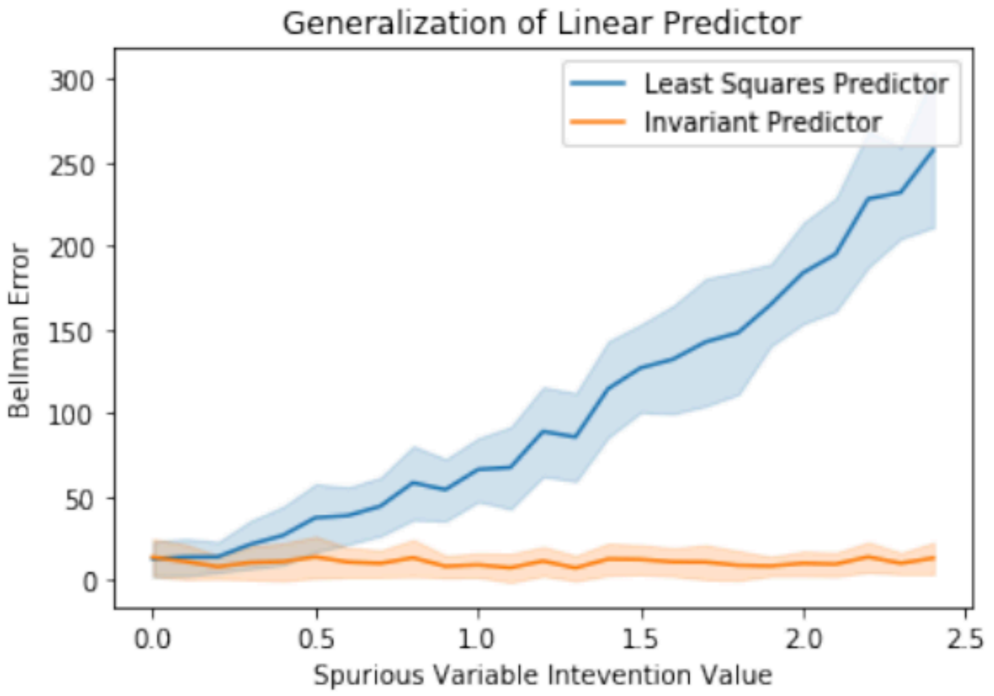
When state is equal to the variables in the causal graph, it's straightforward to apply known causal prediction methods to find the causal ancestors of the reward.

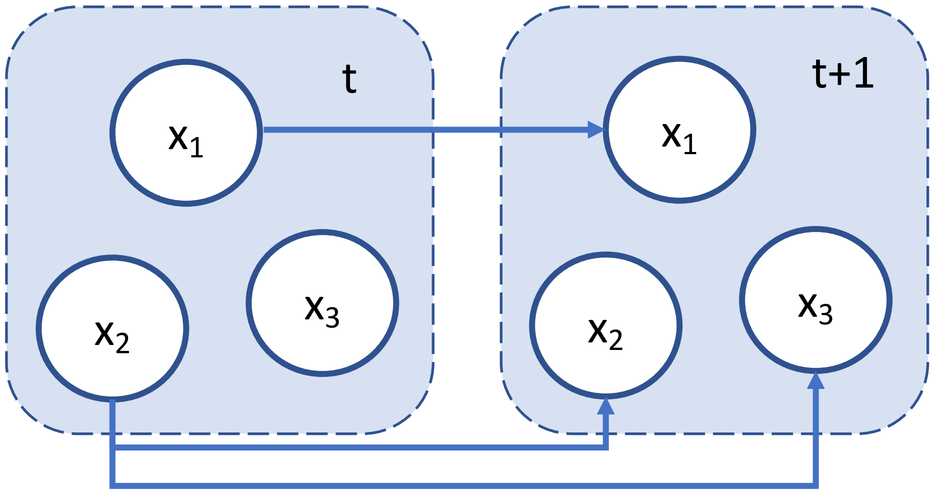
Rich Observation Setting



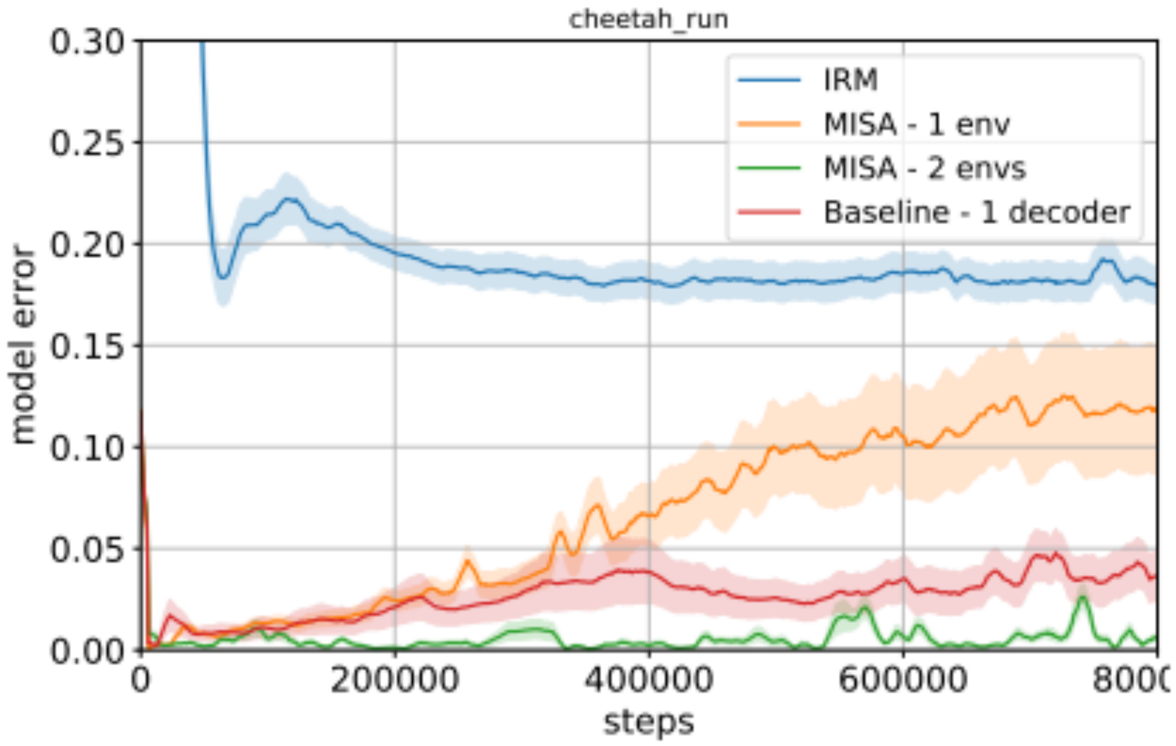
Model Learning in Observable Variables Setting

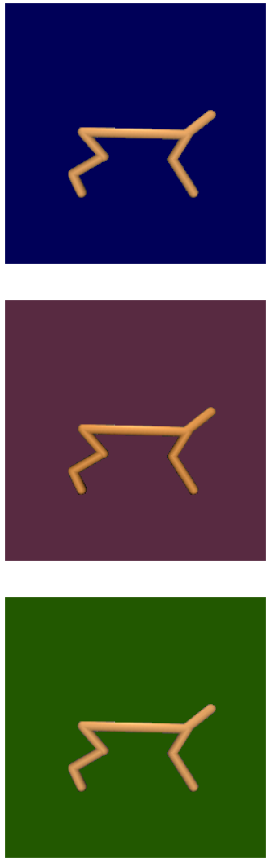
We consider a simple family of MDPs with state space $\mathcal{X} = \{(x_1, x_2, x_3)\}$ with a transition dynamics structure such that $x_1^{t+1} = x_1^t + \epsilon_1^e$, $x_2^{t+1} = x_2^t + \epsilon_2^e$, and $x_3^{t+1} = x_2^t + \epsilon_3^e$



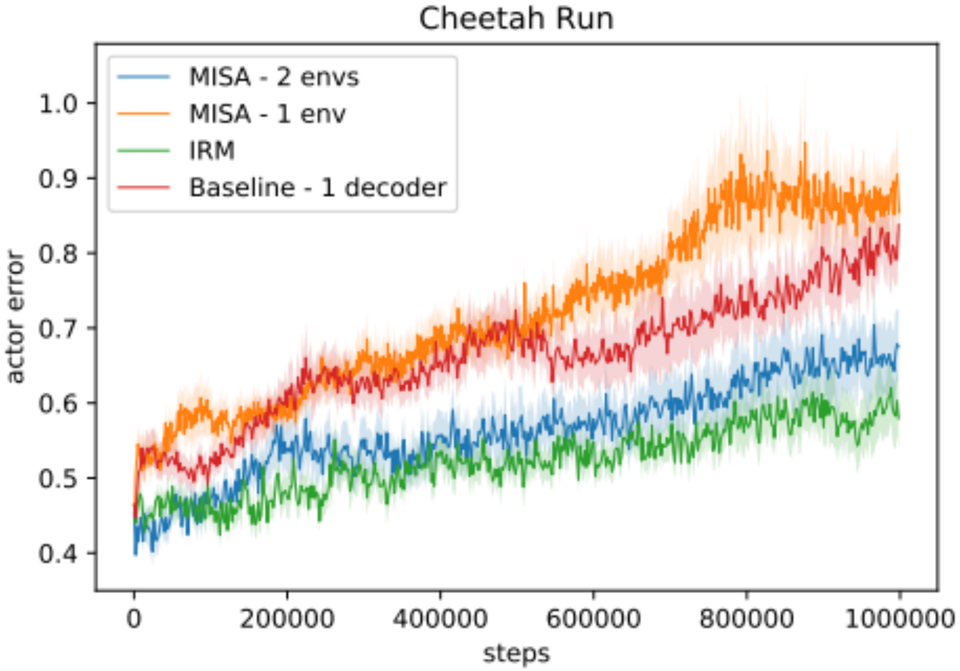


Model Learning in Rich Observation Setting

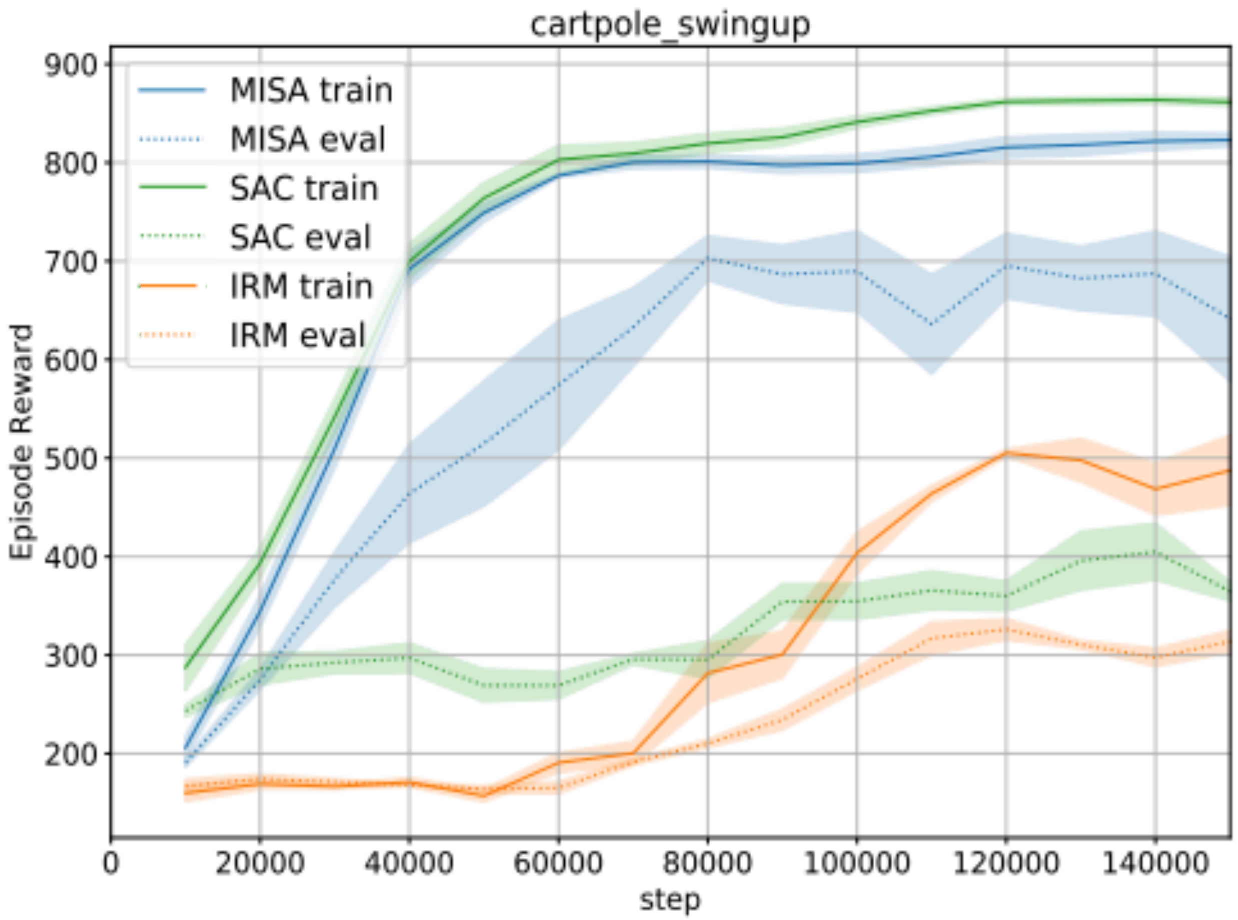




Imitation Learning



Reinforcement Learning



Conclusions

- We show that causal inference methods can be used to find good state abstractions for RL.
- We propose a method to obtain these state abstractions
- We demonstrate that this method works on a variety of deep RL tasks.