Towards Understanding (Cost-Driven) State Representation Learning for Control

Kaiqing Zhang

European Control Conference (ECC) 2024 Tutorial

Learning-Based Control: Fundamentals and Recent Advances

June 26, 2024

Acknowledgement

- Joint work with

Yi Tian (MIT)

Russ Tedrake (MIT)

Suvrit Sra (MIT)

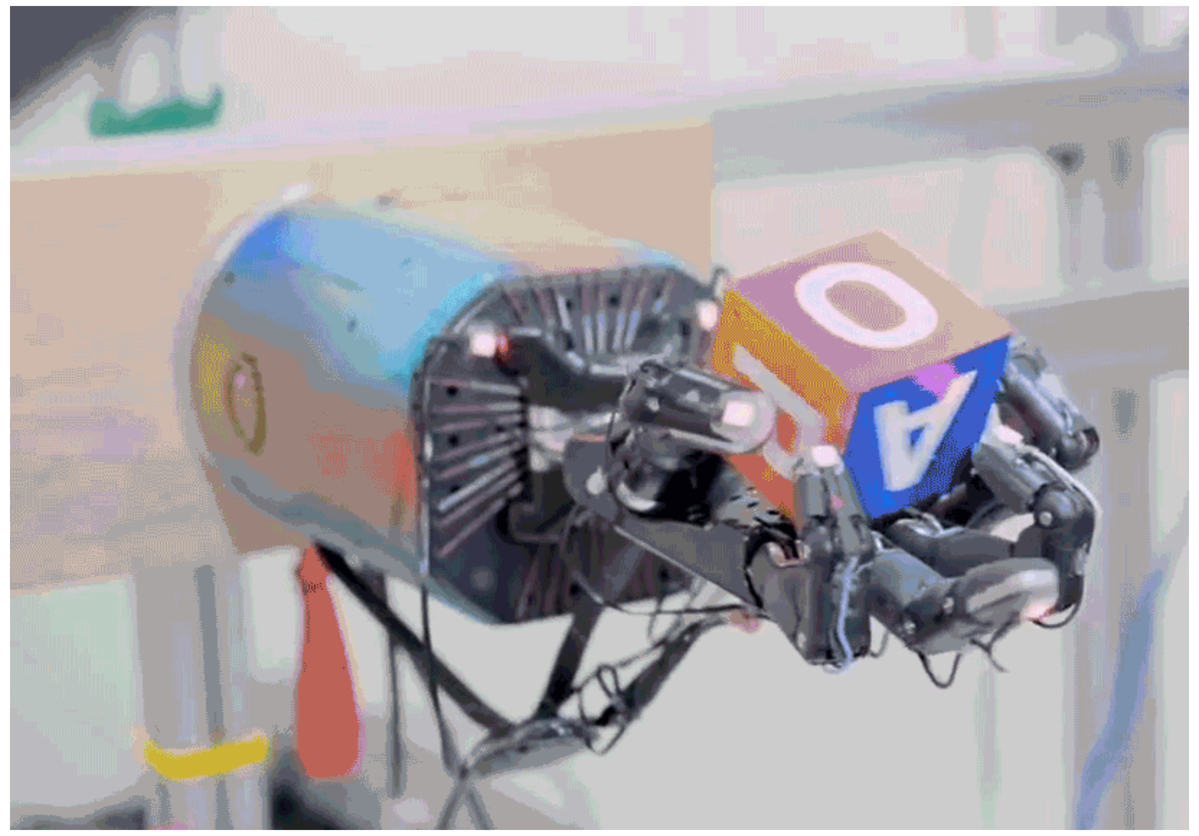
- Also special thanks Alexandre Megretski (MIT) and Horia Mania (Citadel) for helpful discussions

Background

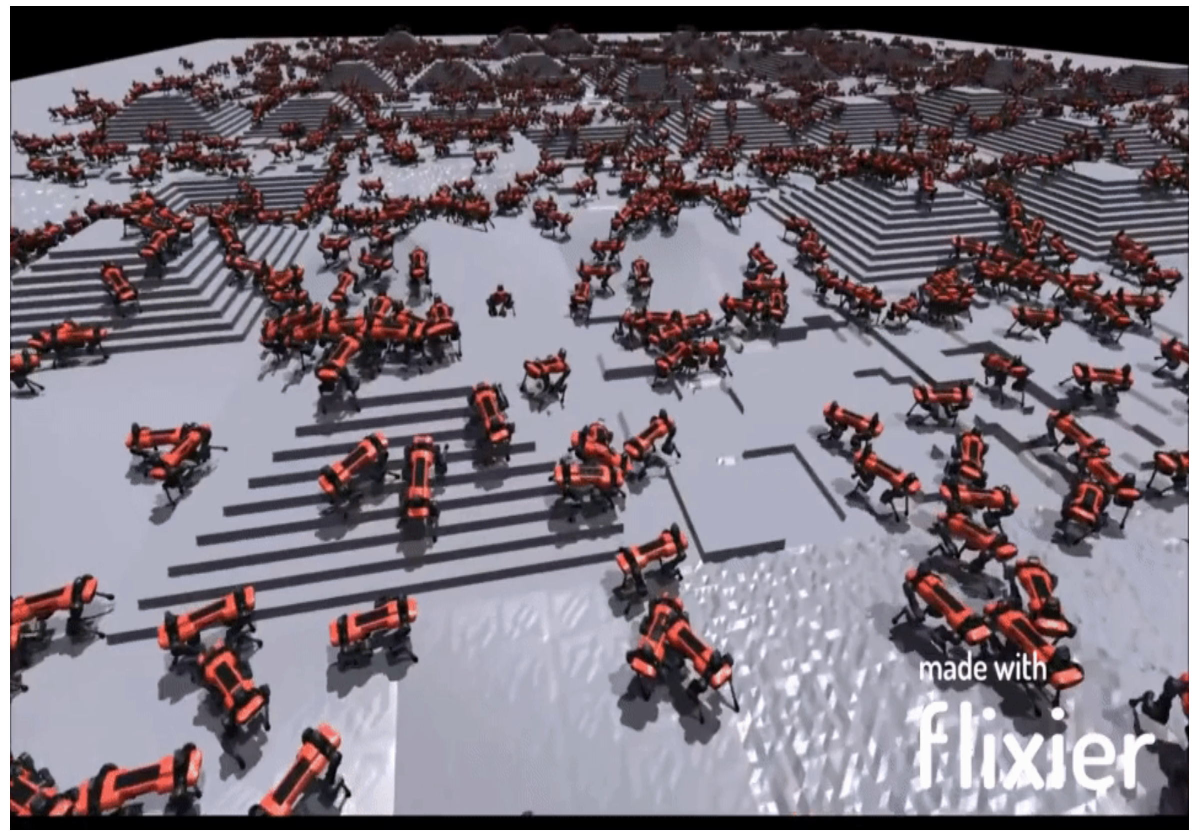
“Robot learning” has fascinated me

- Me entering the Robot Learning (as well as the “Modern RL” world)…

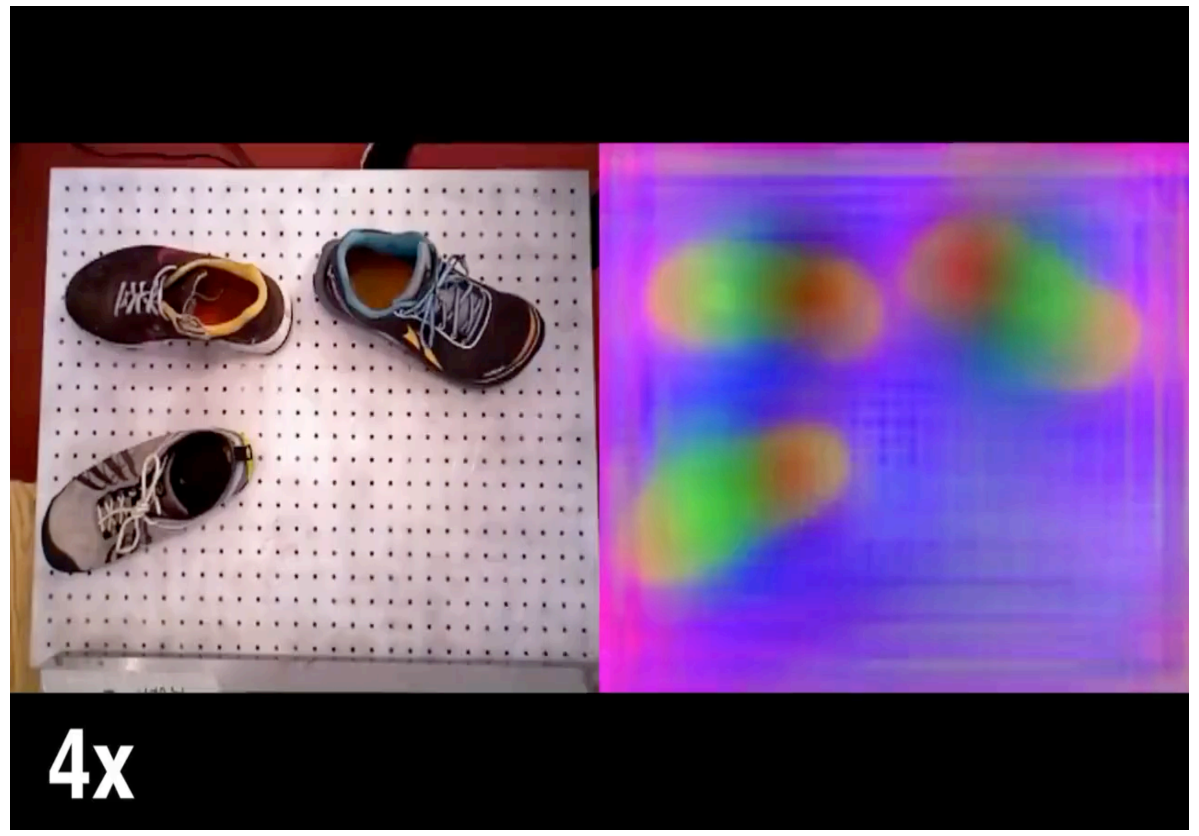
And many many more…



OpenAI, 2018



Marco Hutter’s group at
ETH Zurich, 2021

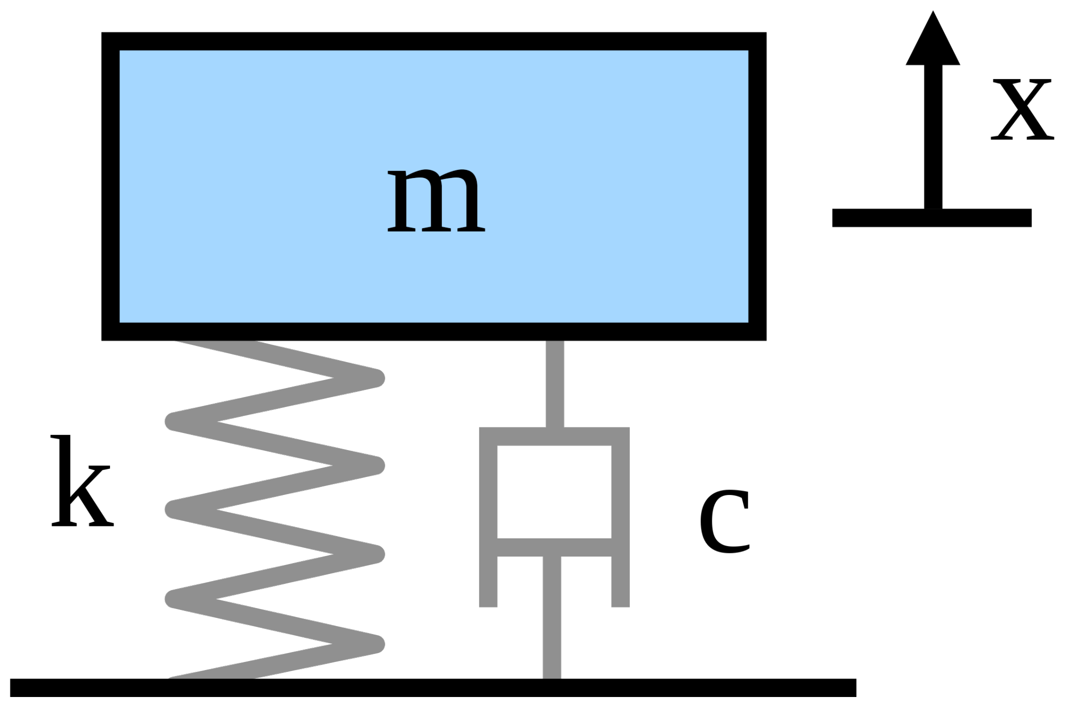


Russ Tedrake’s group, MIT, 2018

Can we try to understand some ideas/principles behind them a bit more?

“State representation” for control

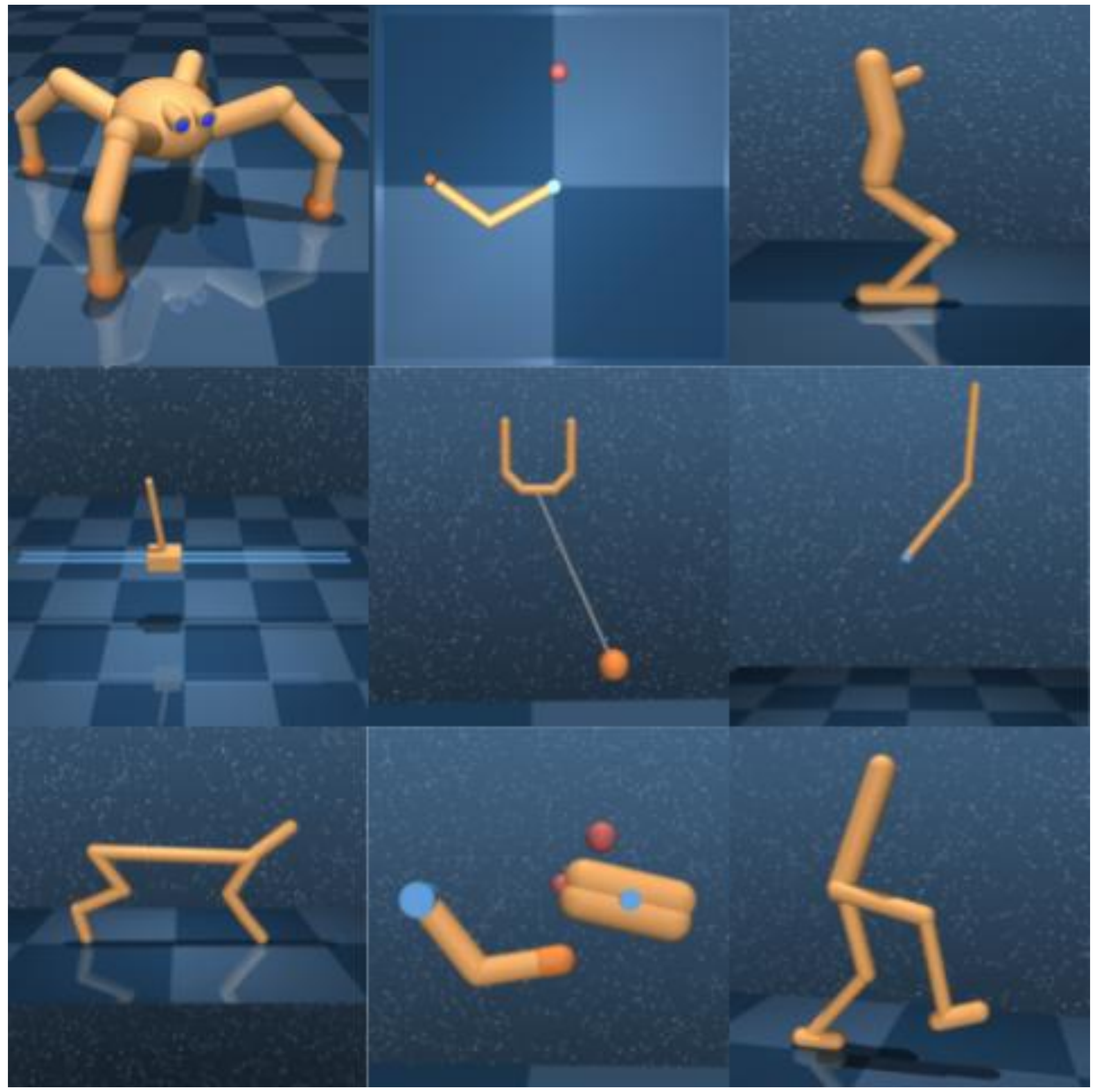
- Control and reinforcement learning (RL) are predominantly based on state-space dynamic models
- In practical (learning for) control systems, e.g., robotic manipulation, the observations, e.g., images, are usually high-dimensional

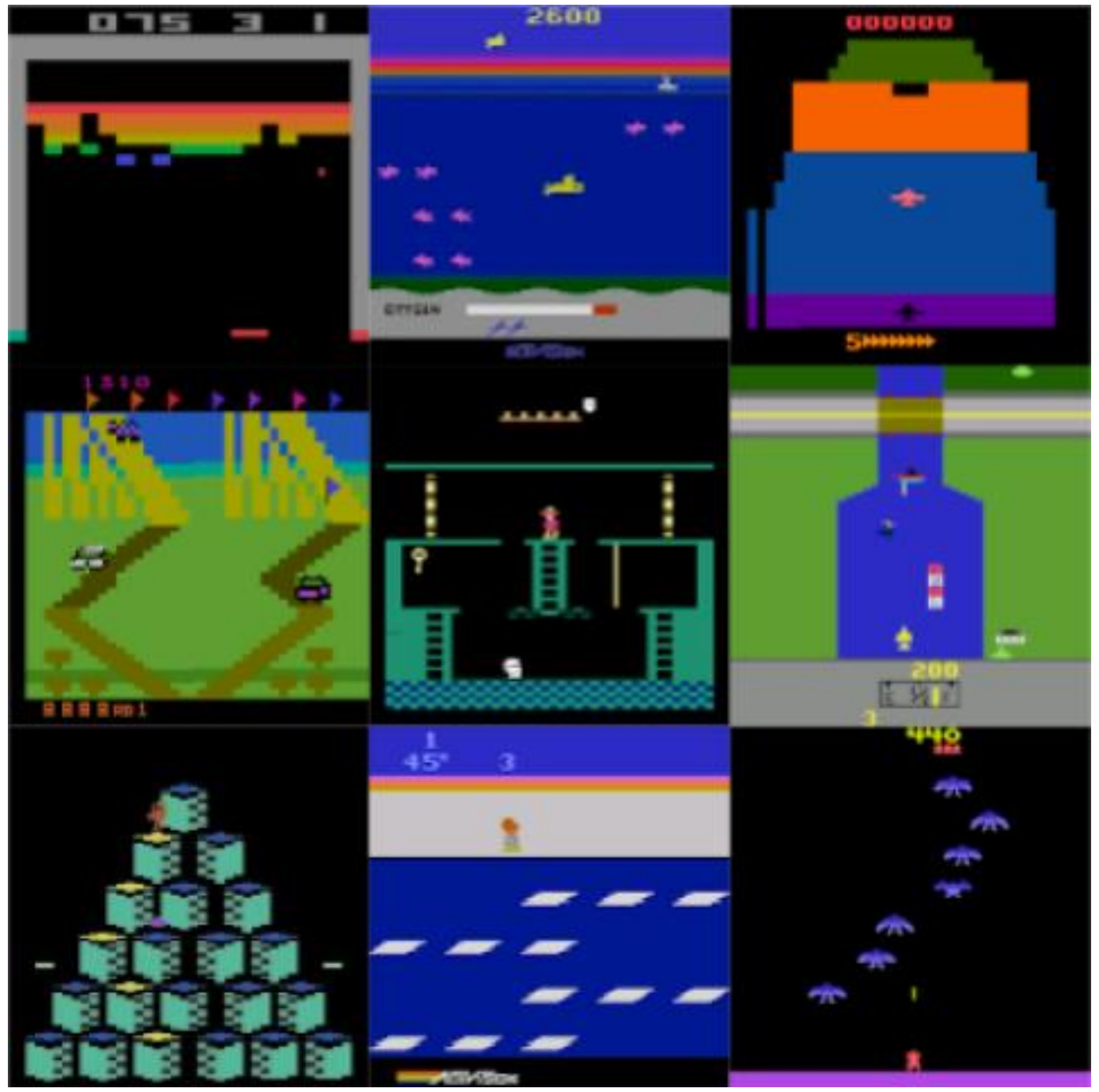


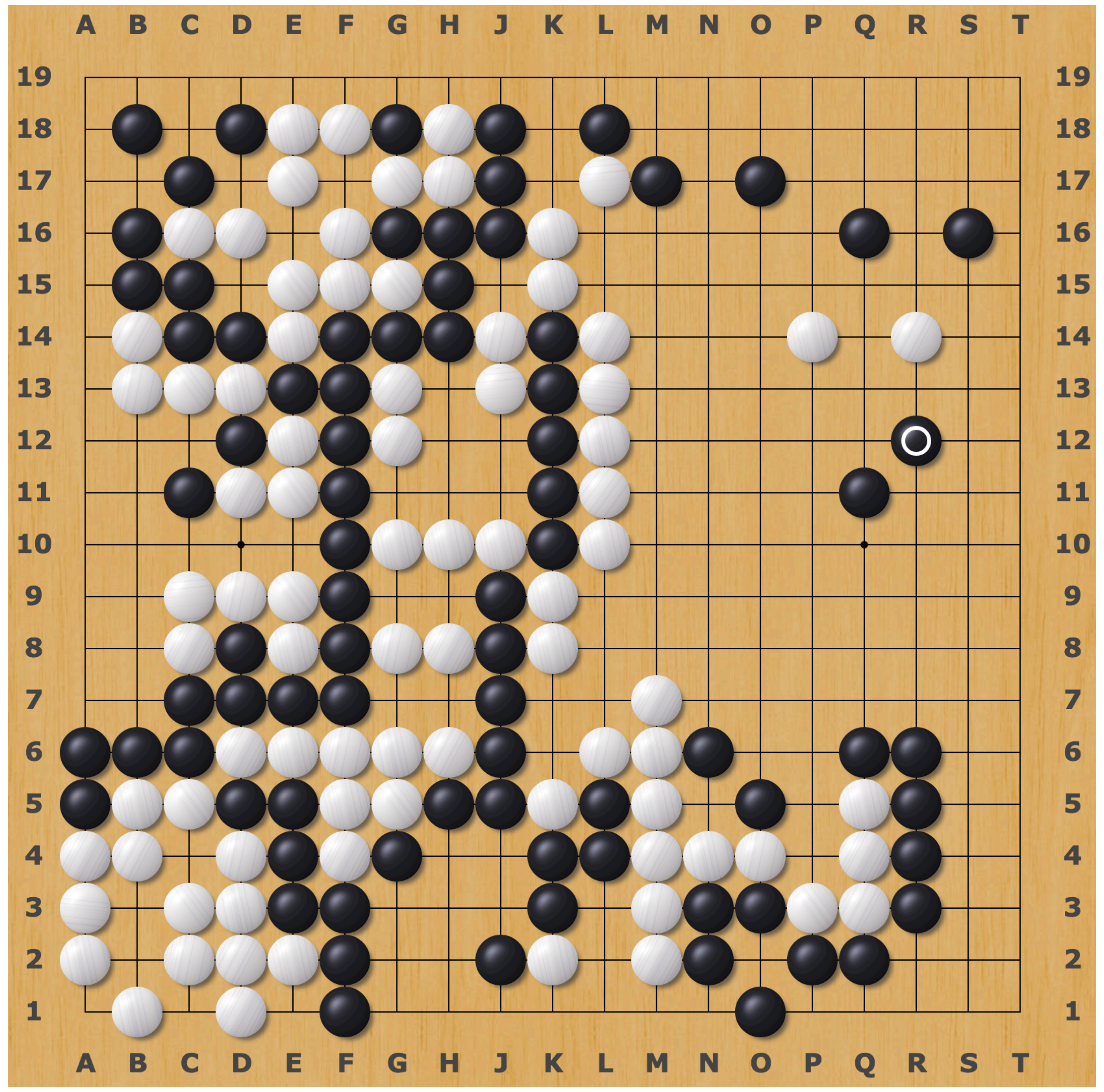
What is a good state (space) and how to learn it from data?

"Latent model learning" for control

- Many empirical works have attempted to learn a latent model for control

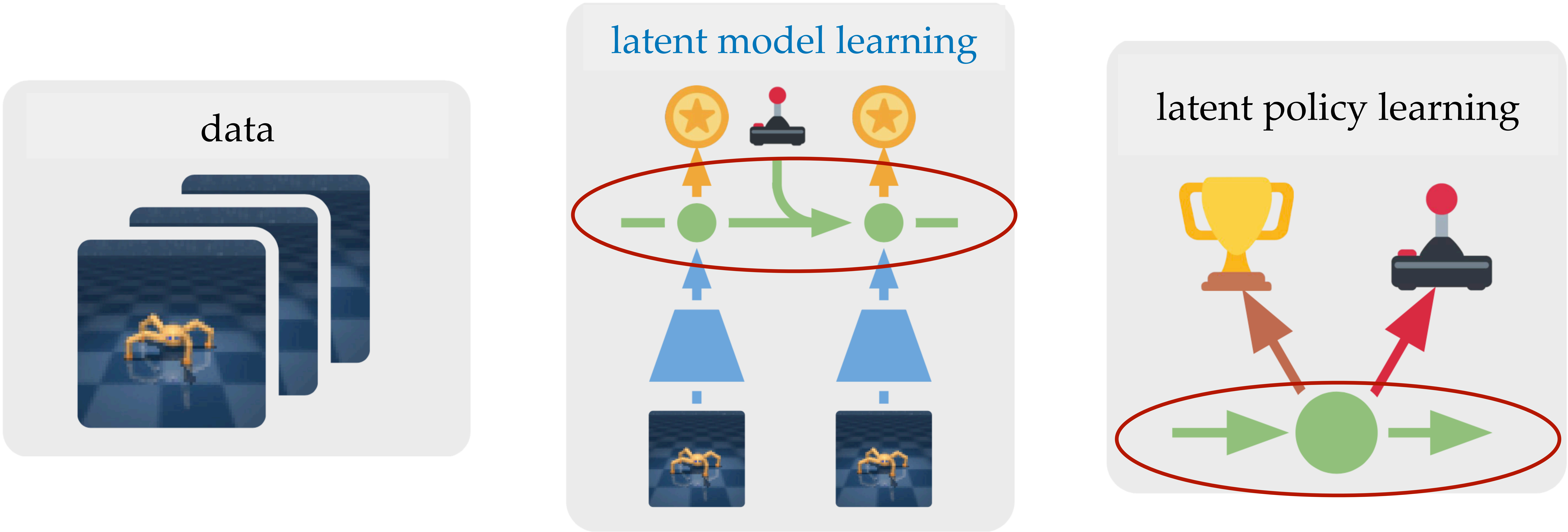






Sources: Left and middle: "Mastering Diverse Domains through World Models." Right: https://online-go.com/.

Latent model learning



Interface with the environment are 3 quantities: **observations**, **actions**, **costs**

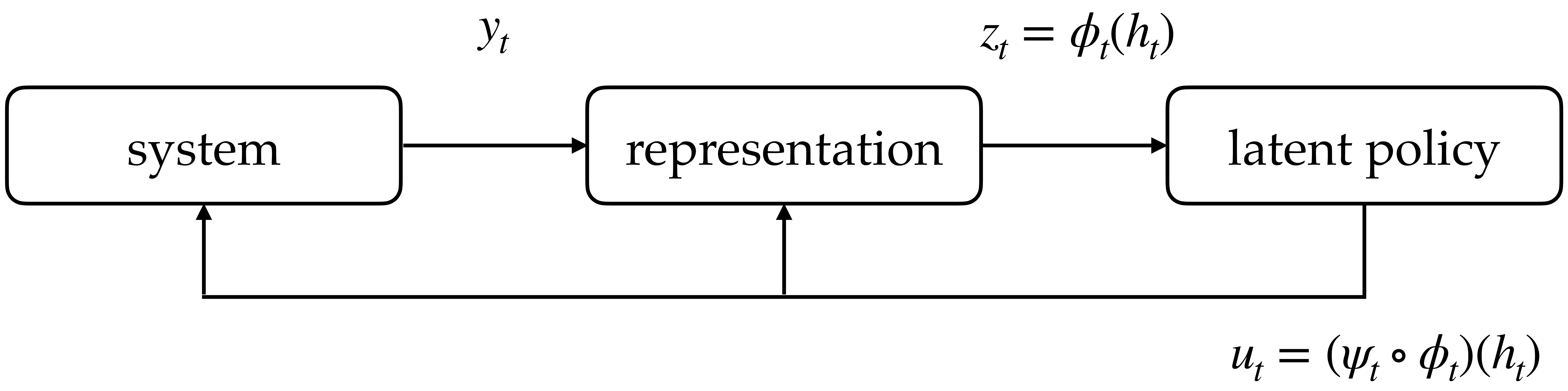
Sources: "Dream to Control: Learning Behaviors by Latent Imagination."

Setup: Control in a partially observable system

- A sequential decision-making problem with time indices $t = 0,1,2,\cdots$
- At step $t \geq 0$, agent observes y_t
- Policy/Controller determines action/control u_t based on history $h_t = (y_0, u_0, \ldots, y_{t-1}, u_{t-1}, y_t)$
- Incur cost c_t at time t
- Finite horizon T, trajectory $(y_0, u_0, c_0, \ldots, y_{T-1}, u_{T-1}, c_{T-1}, y_T, c_T)$
  - Special case: if $\mathbb{P}_x(x_{t+1} \mid x_t, u_t)$ and $\mathbb{P}_y(y_t \mid x_t)$, then it covers partially observed Markov decision processes (POMDP), with broad applications

Anatomy of empirical latent model learning

- Representation function gives latent state by $z_t = \phi_t(z_{t-1}, u_{t-1}, y_t)$ or $z_t = \phi_t(h_t)$
- Latent dynamics $z_{t+1} = f_t(z_t, u_t)$, latent cost $c_t(z_t, u_t) \Rightarrow$ latent policy $\psi_t(u_t \,|\, z_t)$
- Overall policy $(\psi_t \circ \phi_t)_{t=0}^{T-1}$



Motivation

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)

Value Prediction Network

Junhyuk Oh† **Satinder Singh**† **Honglak Lee***,†
†University of Michigan
*Google Brain
{junhyuk,baveja,honglak}@umich.edu, honglak@google.com

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)

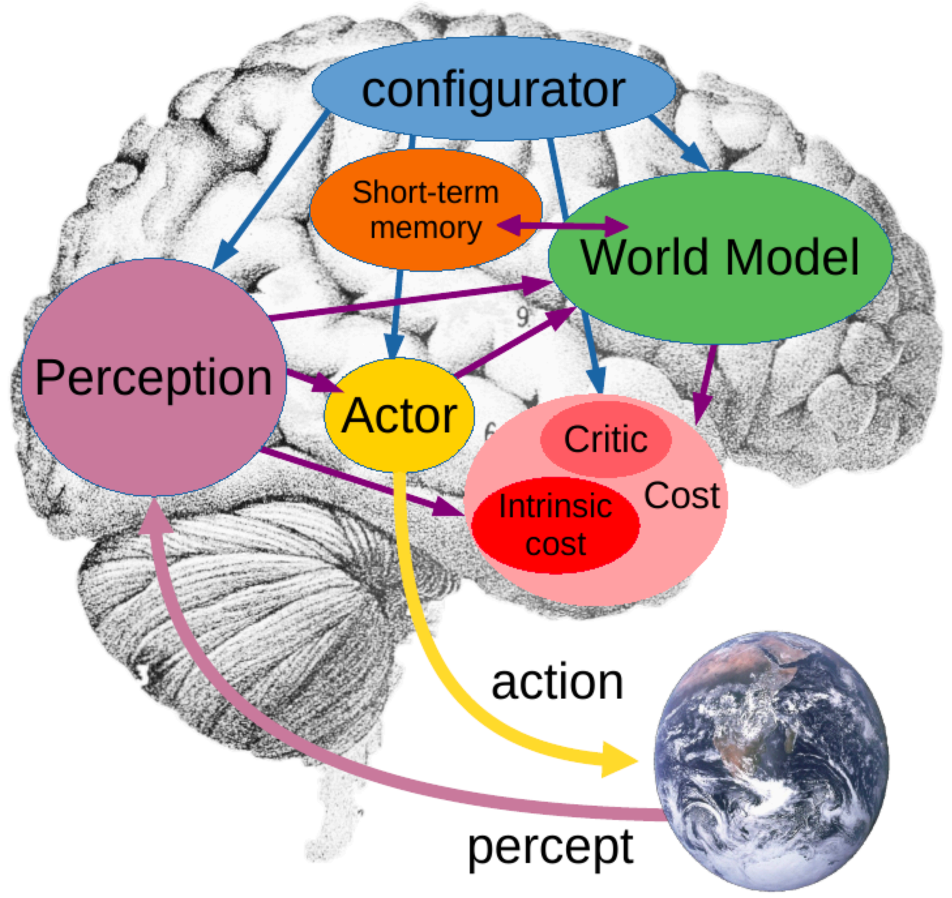
Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak [1] **Pulkit Agrawal** [1] **Alexei A. Efros** [1] **Trevor Darrell** [1]

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)



"A path towards autonomous machine intelligence" — Yann LeCun

---

World Models

David Ha [1] **Jürgen Schmidhuber** [2 3]

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)

Learning Latent Dynamics for Planning from Pixels

Danijar Hafner [1 2] **Timothy Lillicrap** [3] **Ian Fischer** [4] **Ruben Villegas** [1 5]
David Ha [1] **Honglak Lee** [1] **James Davidson** [1]

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)

Mastering Atari, Go, chess and shogi by planning with a learned model

https://doi.org/10.1038/s41586-020-03051-4

Received: 3 April 2020

Accepted: 7 October 2020

Julian Schrittwieser[1,3], Ioannis Antonoglou[1,2,3], Thomas Hubert[1,3], Karen Simonyan[1], Laurent Sifre[1], Simon Schmitt[1], Arthur Guez[1], Edward Lockhart[1], Demis Hassabis[1], Thore Graepel[1,2], Timothy Lillicrap[1] & David Silver[1,2,3] ✉

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)

LEARNING INVARIANT REPRESENTATIONS FOR REINFORCEMENT LEARNING WITHOUT RECONSTRUCTION

Amy Zhang[*12] **Rowan McAllister**[*3] **Roberto Calandra**[2] **Yarin Gal**[4] **Sergey Levine**[3]
[1]McGill University
[2]Facebook AI Research
[3]University of California, Berkeley
[4]OATML group, University of Oxford

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)

Guaranteed Discovery of Control-Endogenous Latent States with Multi-Step Inverse Models

Alex Lamb[*1], Riashat Islam[1,2], Yonathan Efroni[1], Aniket Didolkar[3]
Dipendra Misra[1], Dylan Foster[1], Lekan Molu[1], Rajan Chari[1]
Akshay Krishnamurthy[1], John Langford[*1]
[1] Microsoft Research NYC, New York, USA
[2] School of Computer Science, McGill University, Montreal, Canada
[3] Department of Computer Science, University of Montreal, Montreal, Canada

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)
- Dreamer, DreamerV2, DreamerV3 (Hafner et al., 2020; 2021; 2023)

DREAM TO CONTROL: LEARNING BEHAVIORS BY LATENT IMAGINATION

Danijar Hafner *
University of Toronto
Google Brain

Timothy Lillicrap
DeepMind

Jimmy Ba
University of Toronto

Mohammad Norouzi
Google Brain

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)
- Dreamer, DreamerV2, DreamerV3 (Hafner et al., 2020; 2021; 2023)

MASTERING ATARI WITH DISCRETE WORLD MODELS

Danijar Hafner * Google Research
Timothy Lillicrap DeepMind
Mohammad Norouzi Google Research
Jimmy Ba University of Toronto

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)
- Dreamer, DreamerV2, DreamerV3 (Hafner et al., 2020; 2021; 2023)

Mastering Diverse Domains through World Models

Danijar Hafner[1,2] Jurgis Pasukonis[1] Jimmy Ba[2] Timothy Lillicrap[1]

[1]DeepMind [2]University of Toronto

Empirical latent model learning methods

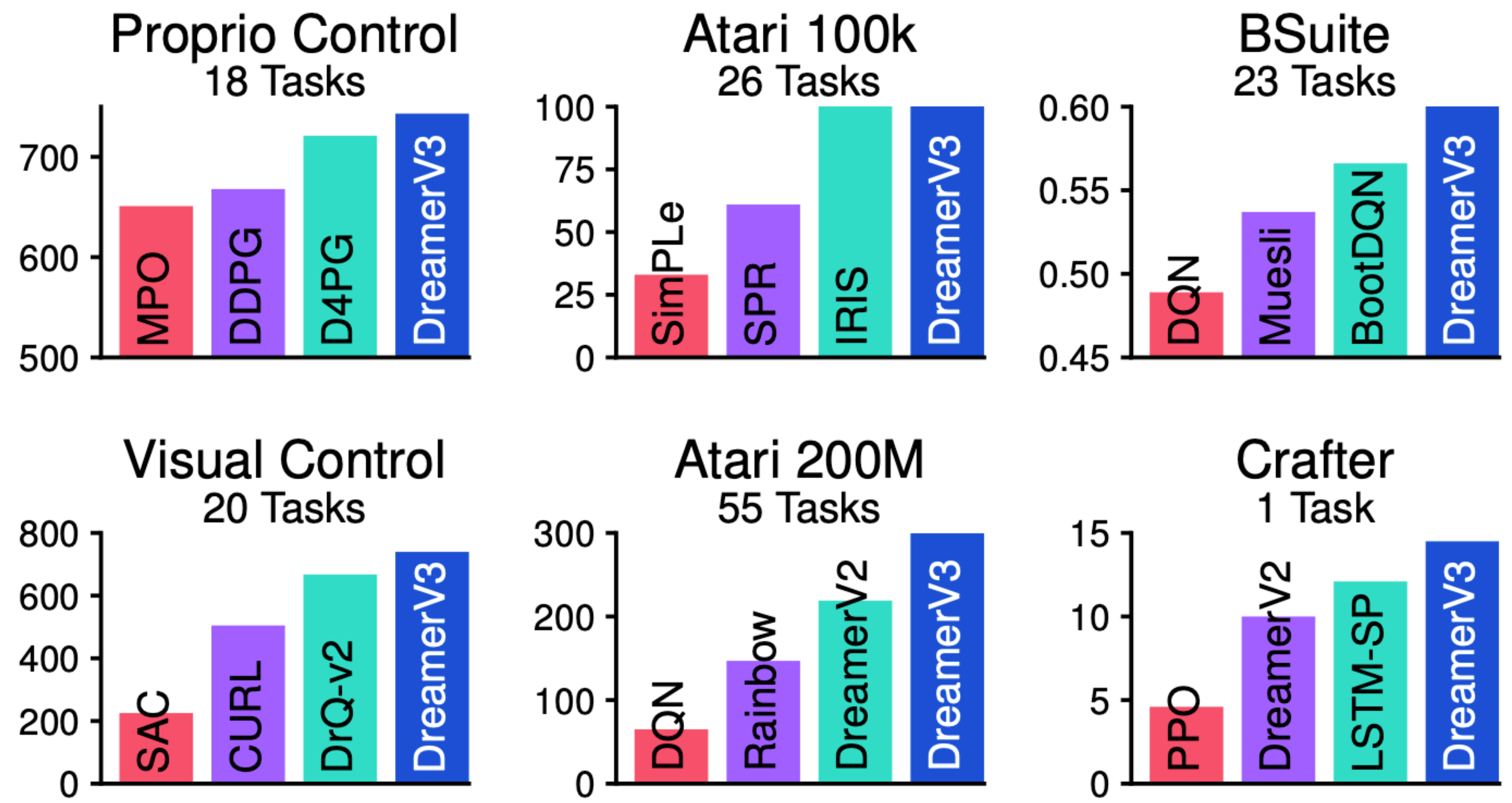
Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)
- Dreamer, DreamerV2, DreamerV3 (Hafner et al., 2020; 2021; 2023)
- …

Empirical latent model learning methods

Many empirical works have attempted to learn a latent model for control

- Value Prediction Network (Oh et al., 2017)
- Self-Supervised Prediction (Pathak et al., 2017)
- World Models (Ha and Schmidhuber, 2018)
- PlaNet (Hafner et al., 2019)
- MuZero (Schrittwieser et al., 2020)
- Deep Bisimulation (Zhang et al., 2021)
- AC-State (Lamb et al., 2022)
- Dreamer, DreamerV2, DreamerV3 (Hafner et al., 2020; 2021; 2023)
- …



Motivation of this work

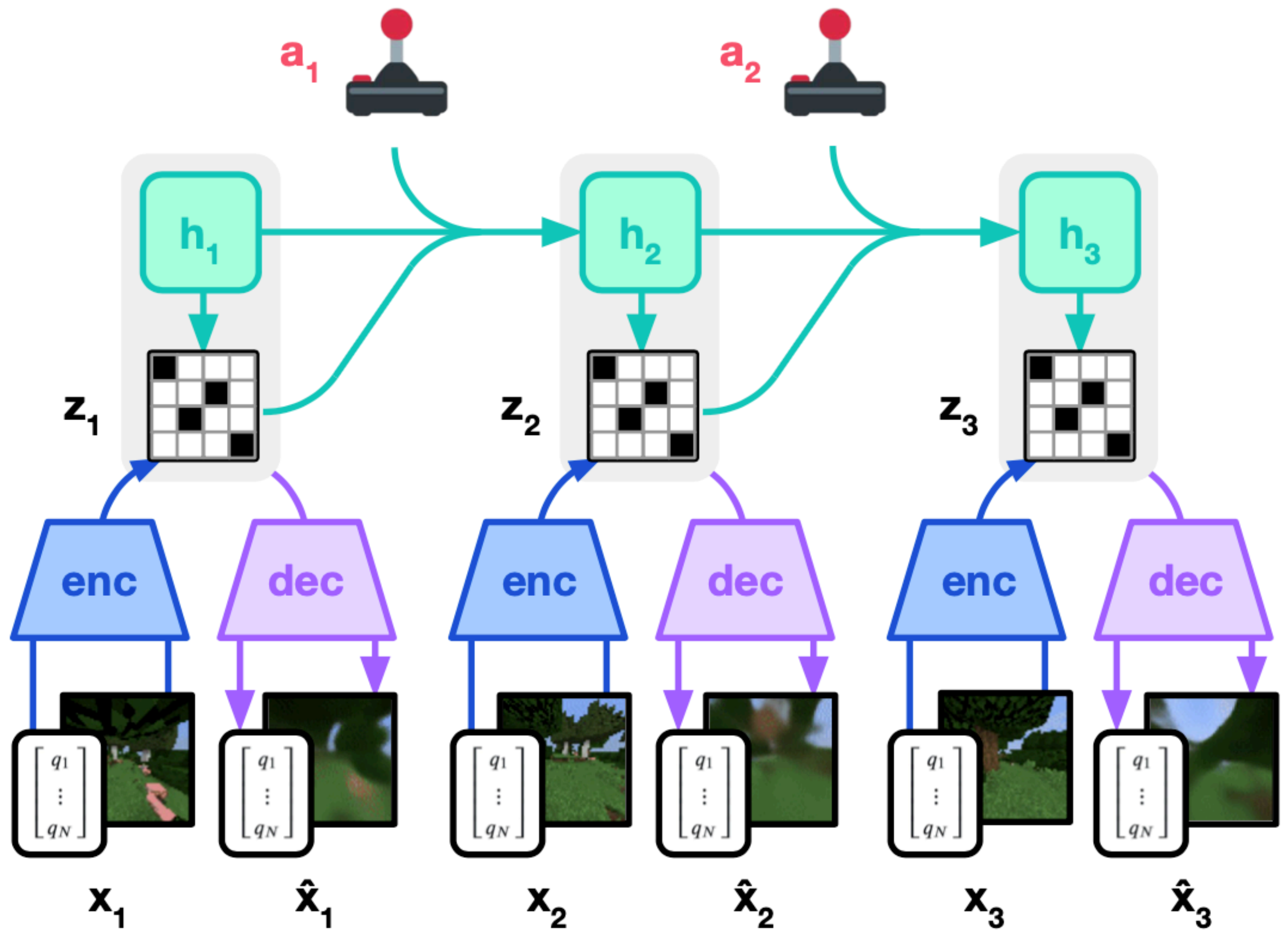
- Despite exciting empirical advances, theoretical understanding is relatively lacking
- What (latent state spaces) are these empirical methods essentially learning, with a finite-number of samples?
- Even for very basic partially observable control systems, the answer was unknown
  - Should pass the sanity-check for these basic control systems?
  - Gain some insights from basic control problems

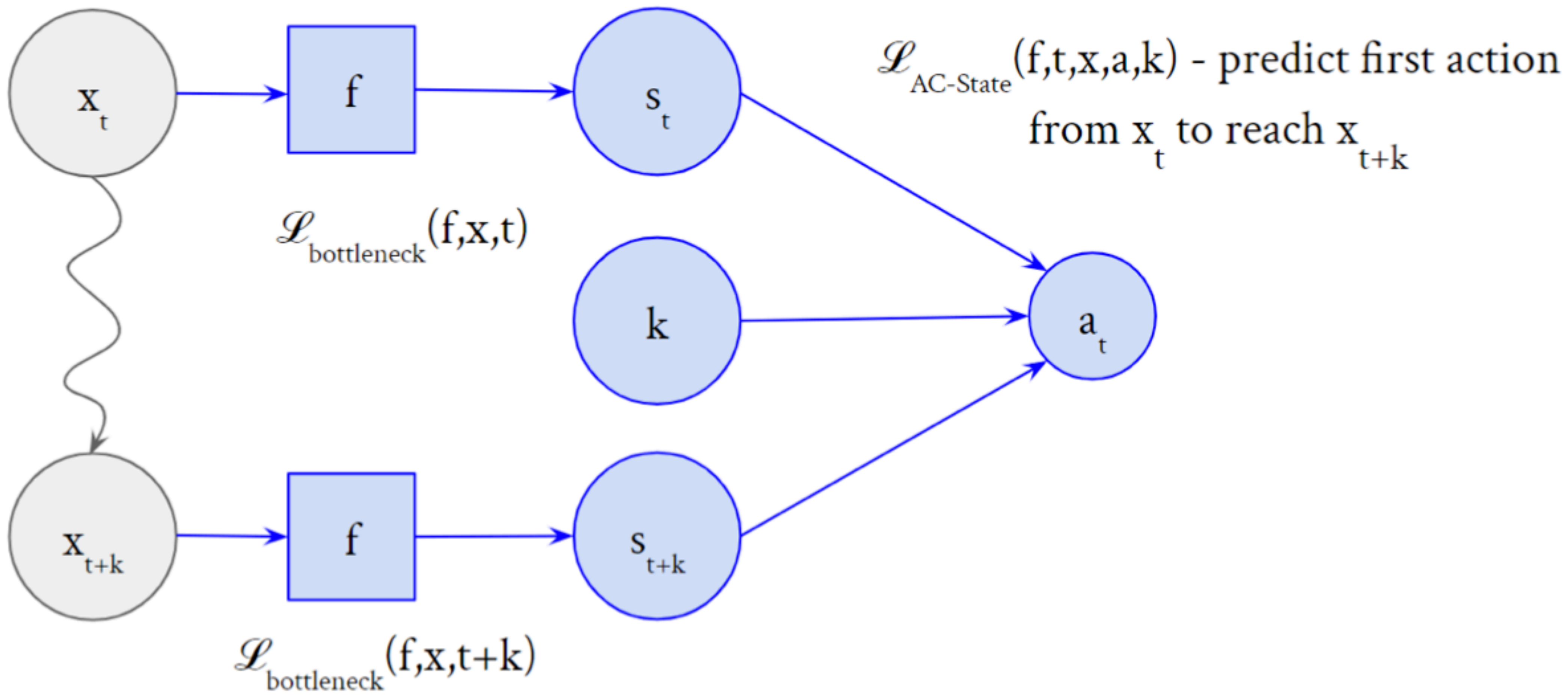
Motivation of this work

- Higher-level: What's the minimal condition / right objective for latent model learning that works for downstream control tasks? 3 quantities: **observations**, **actions**, **costs**

Motivation of this work

- Higher-level: What's the minimal condition / right objective for latent model learning that works for downstream control tasks? 3 quantities: **observations**, **actions**, **costs**
  - Observation-driven: Reconstructing Observation — World Models (Ha and Schmidhuber, 2018), PlaNet and the Dreamer series (Hafner et al., 2019; 2020; 2021; 2023), etc.
  - Action-driven: Inverse Models — (Pathak et al., 2017), AC-State (Lamb et al., 2022), etc.

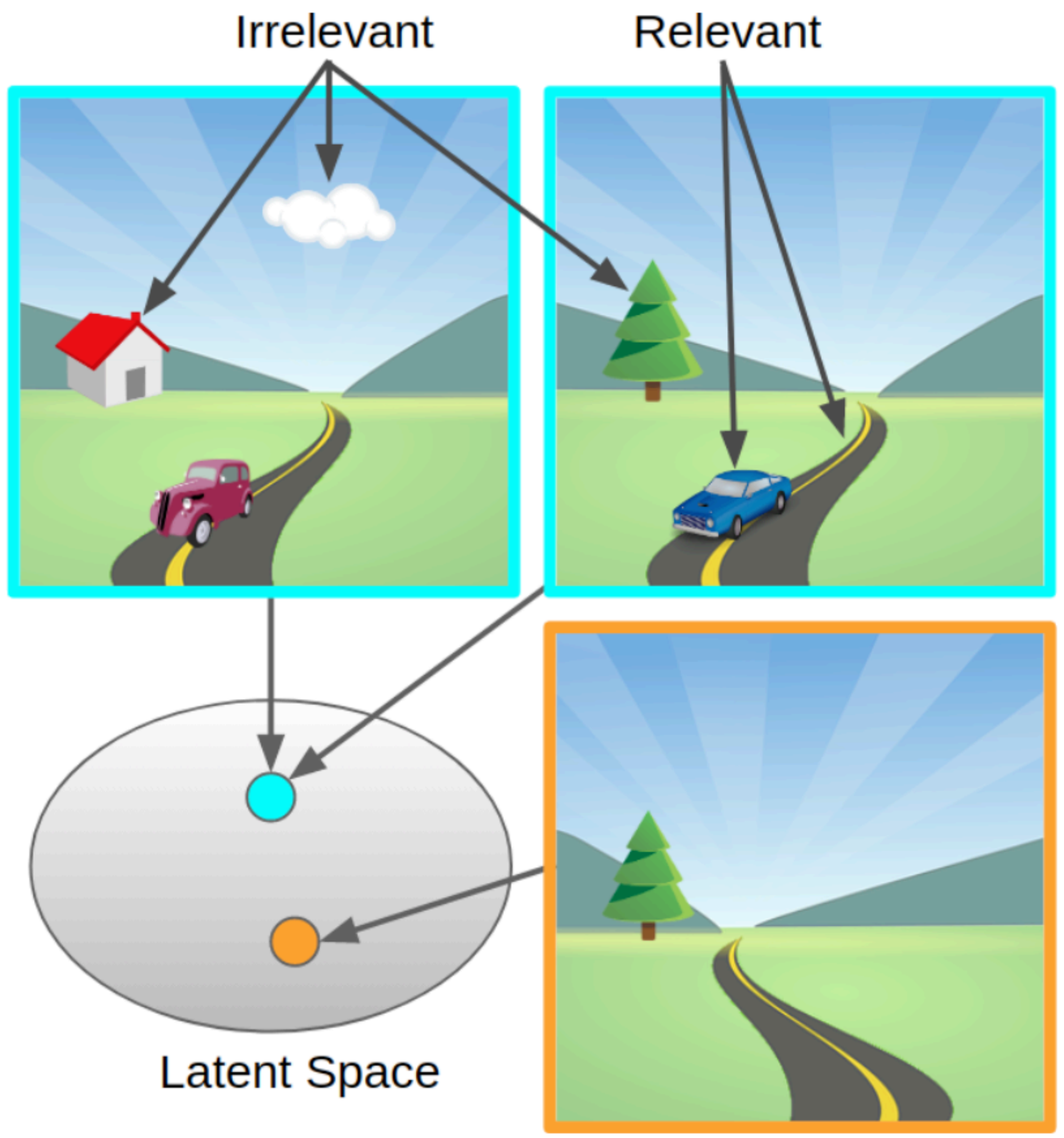




Source: Left: "Mastering Diverse Domains through World Models ". Right: "Guaranteed Discovery of Control-Endogenous Latent States with Multi-Step Inverse Models ".

Motivation of this work

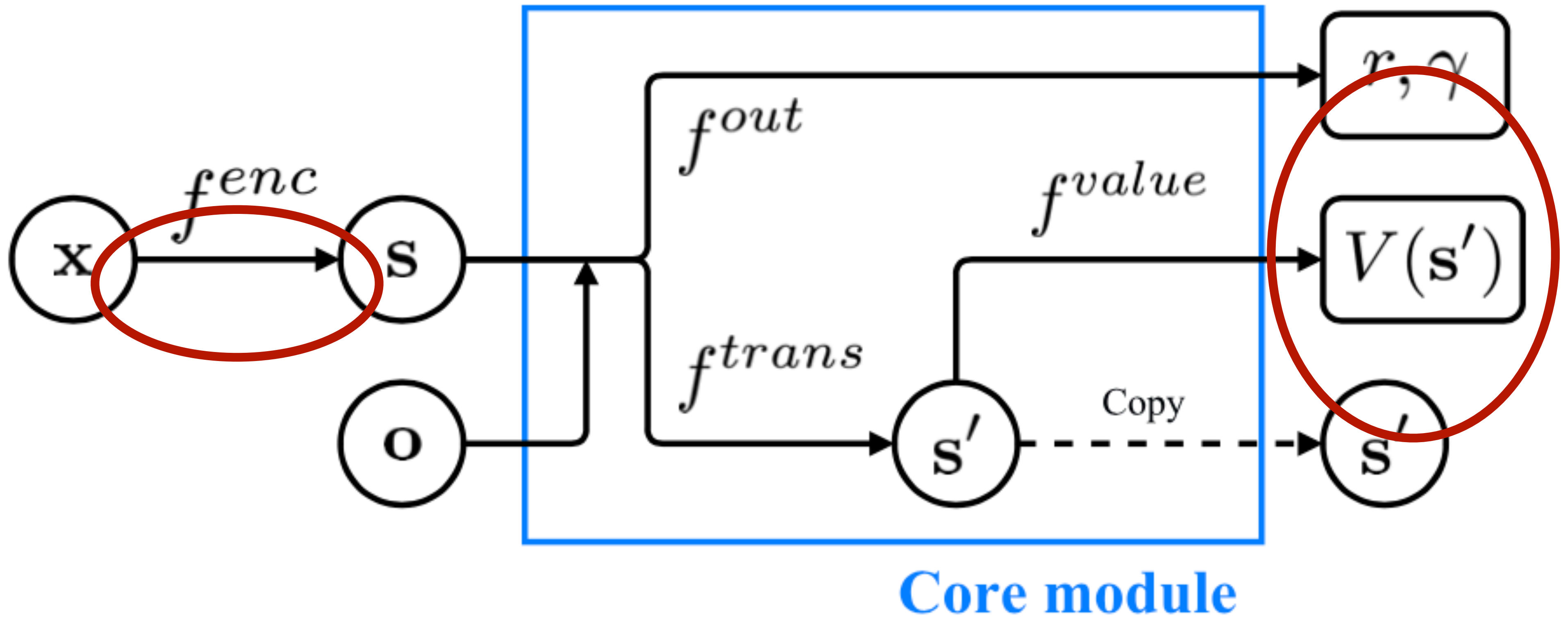
- Minimal condition/Right objective for latent model learning that works for control?
  - Objectives in reconstructing observation and inverse models are task-agnostic
    - Pros: Can be universal and multi-task/generalizable
    - Cons: May contain control-irrelevant information
    - Cons: Easily distracted by noises
    - Cons: Obs. can be high-dimensional and hard to predict



Source: "Learning Invariant Representations for Reinforcement Learning without Reconstruction".

Motivation of this work

- Minimal condition/Right objective for latent model learning that works for control?
  - Objectives in reconstructing observation and inverse models are task-agnostic
  - Cost-driven: (Cumulative) cost prediction — Value Prediction Network (Oh et al., 2017), MuZero (Schrittwieser et al., 2020), Deep Bisimulation (Zhang et al., 2021), etc.
    - It is task-specific, necessary for planning, and thus more "direct"



Source: "Value Prediction Network".

Motivation of this work

- Minimal condition/Right objective for latent model learning that works for control?
  - Objectives in reconstructing observation and inverse models are task-agnostic
  - Cost-driven: (Cumulative) cost prediction — Value Prediction Network (Oh et al., 2017), MuZero (Schrittwieser et al., 2020), Deep Bisimulation (Zhang et al., 2021), etc.
    - It is task-specific, necessary for planning, and thus more "direct"

Can cost-driven direct latent model learning provably solve partially observable control?

Problem Formulation

Linear-quadratic-Gaussian control (LQG)

- Linear time-varying (LTV) model of LQG: for $t \geq 0$,

$$x_{t+1} = A_t x_t + B_t u_t + w_t,$$
$$y_t = C_t x_t + v_t,$$

where $w_t \sim \mathcal{N}(0,\Sigma_{w_t})$, $v_t \sim \mathcal{N}(0,\Sigma_{v_t})$ are i.i.d. Gaussian and initial state $x_0 \sim \mathcal{N}(0,\Sigma_0)$

- Cost $c_t(x, u) = \|x\|^2_{Q_t} + \|u\|^2_{R_t}$, terminal cost $c_T(x) = \|x\|^2_{Q_T}$

$$\boxed{\text{Goal:} \qquad \min_{\pi} \quad J^\pi = \mathbb{E}^\pi\Big[\sum_{t=0}^{T} c_t\Big]}$$

- If model is known: optimal control is the Kalman filter (one type of latent model!)

$$z_0 = L_0 y_0, \quad z_{t+1} = A_t z_t + B_t u_t + L_{t+1}(y_{t+1} - C_{t+1}(A_t z_t + B_t u_t))$$
$$= \overline{A}_t z_t + \overline{B}_t u_t + L_{t+1} y_{t+1},$$

combined with a linear-quadratic regulator $u_t = K_t z_t$, where $(L_t, K_t)_{t=0}^{T-1}$ are given by Riccati difference equations

Theoretical works on *learning* LQG: Sys-ID

- For unknown time-invariant LQG, "standard" treatment for finite-sample analysis lately (Oymak and Ozay, 2018; Simchowitz et al., 2019; Lale et al., 2021; Zheng & Li, 2021) uses Markov parameters for system identification (Ljung, 1998)

Theoretical works on *learning* LQG: Sys-ID

- For unknown time-invariant LQG, "standard" treatment for finite-sample analysis lately (Oymak and Ozay, 2018; Simchowitz et al., 2019; Lale et al., 2021; Zheng & Li, 2021) uses Markov parameters for system identification (Ljung, 1998)

- The Markov parameter maps control actions to observations

$$y_t = \underbrace{[0, CB, CAB, \ldots, CA^{\tau-2}B]}_{\text{Markov parameter}}[u_t; u_{t-1}; \ldots; u_{t-\tau+1}] + \underbrace{CA^{\tau-1}x_{t-\tau+1}}_{\text{decay to zero}}$$

- Once the Markov parameter is learned, (A, B, C) are recovered by the Ho-Kalman algorithm

Problems?

- This pipeline is specific to linear (time-invariant) systems
- Learning Markov parameters is still "reconstructing" observation

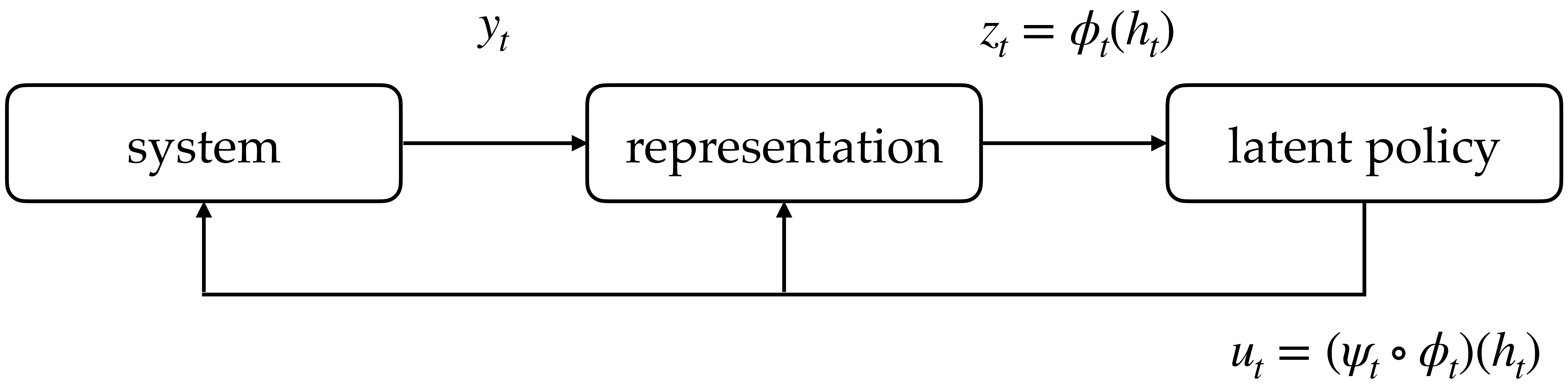
Our Approach

Recall: Anatomy of empirical latent model learning

- Representation function gives latent state by $z_t = \phi_t(z_{t-1}, u_{t-1}, y_t)$ or $z_t = \phi_t(h_t)$
- Latent dynamics $z_{t+1} = f_t(z_t, u_t)$
- Latent cost $c_t(z_t, u_t)$

$\Rightarrow$ latent policy $\psi_t(u_t \,|\, z_t)$

- Overall policy $(\psi_t \circ \phi_t)_{t=0}^{T-1}$

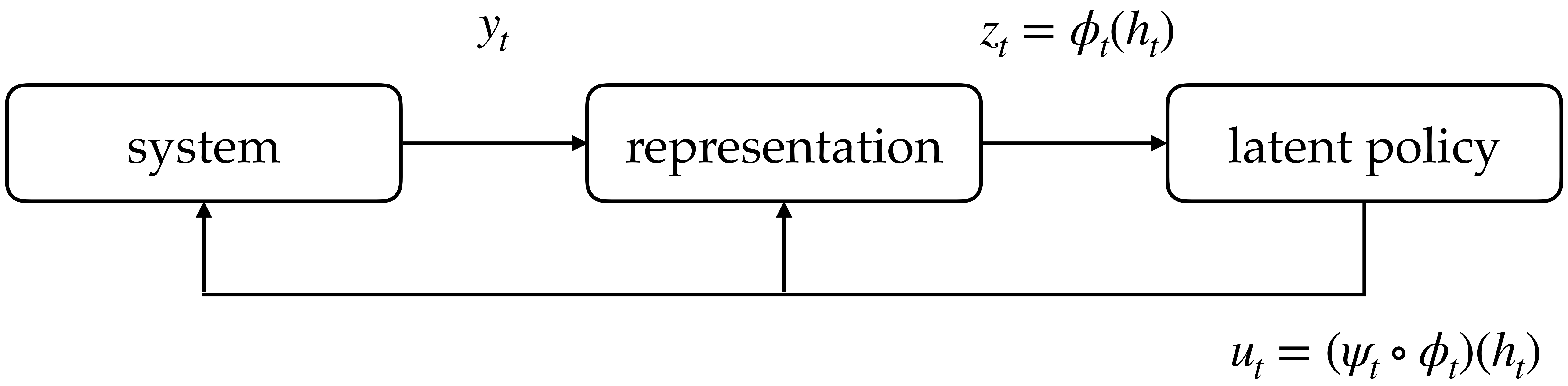


Cost-driven latent model learning for LQG

- Representation function gives latent state by $z_t = M_t h_t$ or $z_t = \overline{A}_{t-1} z_{t-1} + \overline{B}_{t-1} u_{t-1} + L_t y_t$
- Latent dynamics $z_{t+1} = A_t z_t + B_t u_t$
- Latent cost $c_t(z_t, u_t) = \|z_t\|_{Q_t}^2 + \|u_t\|_{R_t}^2$

$$\left.\begin{array}{l} z_{t+1} = A_t z_t + B_t u_t \\ c_t(z_t, u_t) = \|z_t\|_{Q_t}^2 + \|u_t\|_{R_t}^2 \end{array}\right\} \Rightarrow \text{latent policy } u_t = K_t z_t$$

- Overall policy $(M_t, K_t)_{t=0}^{T-1}$ or $L_0, (\overline{A}_t, \overline{B}_t, L_t, K_t)_{t=0}^{T-1}$



Cost-driven latent model learning for LQG

- Representation function gives latent state by $z_t = M_t h_t$
- Latent dynamics $z_{t+1} = A_t z_t + B_t u_t$
- Latent cost $c_t(z_t, u_t) = \|z_t\|_{Q_t}^2 + \|u_t\|_{R_t}^2$
- Overall policy $(M_t, K_t)_{t=0}^{T-1}$

Cost-driven latent model learning:

Given n trajectories, solve

$$\min_{M_t, Q_t, R_t, b_t} \quad \sum_{t=0}^{T} \sum_{i=1}^{n} \underbrace{(\|M_t h_t^{(i)}\|_{Q_t}^2 + \|u_t^{(i)}\|_{R_t}^2 + b_t - c_t^{(i)})^2}_{\textbf{cost prediction error}}$$

Cost-driven latent model learning for LQG

Main difference to World Model (Ha and Schmidhuber, 20): they did observation-reconstruction-based approach with autoencoder

1. Data collection: n trajectories using actions $u_t \sim \mathcal{N}(0, \sigma_u^2 I)$

2. State representation learning: find the M_t by solving

$$\min_{M_t, b_t} \sum_{i=1}^{n} \left(\|M_t h_t^{(i)}\|^2 + \sum_{\tau=t}^{t+m-1} \|u_\tau^{(i)}\|_{R_t}^2 + b_t - \sum_{\tau=t}^{t+m-1} c_t^{(i)} \right)^2$$

3. Latent dynamics learning: convert history to latent state by $z_t^{(i)} = M_t h_t^{(i)}$ and use $(z_t^{(i)}, u_t^{(i)}, z_{t+1}^{(i)}, c_t^{(i)})$ to identify latent model parameters A_t, B_t by ordinary linear regression

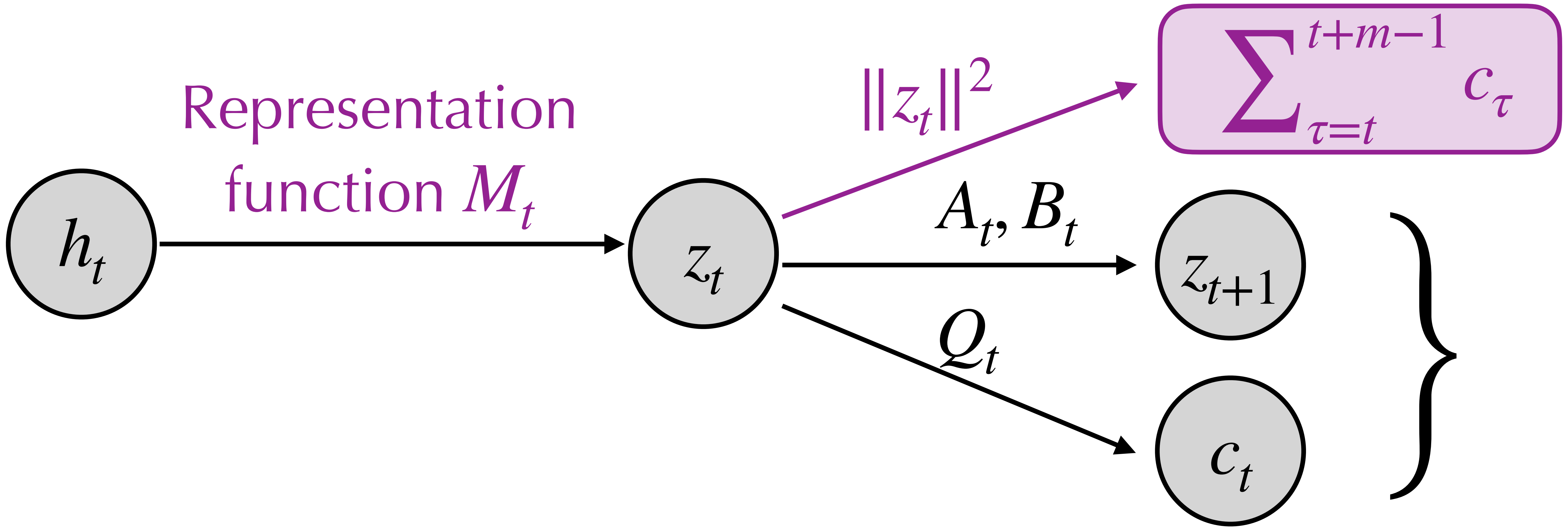
4. Latent policy computation: apply the Riccati difference equation to compute feedback gain K_t

5. Return policy $(K_t \circ M_t)_{t=0}^{T-1}$

Main modifications to previous cost: cumulative cost (step 2)

Cost-driven latent model learning for LQG

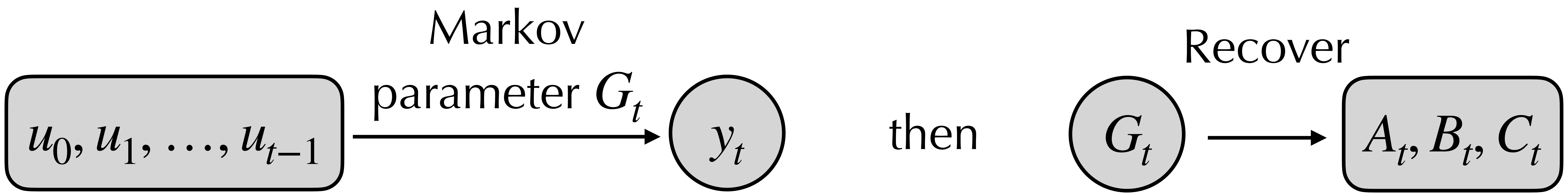
Cost-driven latent model learning (ours)



Step 1: learn M_t by cumulative cost prediction

Step 2: identify A_t, B_t, Q_t in the latent system

Classical system identification (Sys-ID)



Main Results

Main Results

Can cost-driven latent model learning solve LQG control?

Theorem. Given an unknown LQG control problem with horizon T, under standard assumptions including stability, controllability (within ℓ steps) and cost observability, cost-driven latent model learning returns from n collected trajectories, with high probability (hiding poly dependence on prob. parameters)

- A state representation function that is $\tilde{\mathcal{O}}(\ell^{1/2} n^{-1/4})$-optimal in the first ℓ steps and $\tilde{\mathcal{O}}(T^{3/2} n^{-1/2})$-optimal in the next $T-\ell$ steps;
- A latent policy that is $\tilde{\mathcal{O}}((\mathcal{O}(1))^{\ell} \ell n^{-1/4})$-optimal in the first ℓ steps and $\tilde{\mathcal{O}}(T^4 n^{-1})$-optimal in the next $T-\ell$ steps.

Remarks & Challenges

- For LQG control, (cumulative) scalar cost is informative to recover the near-optimal state representation function
  - The insight of predicting cumulative cost in latent model learning has also been empirically observed in MuZero (Schrittwieser et al., 2020)
- Challenge 1: Matrix quadratic regression in cost prediction — covariates are product of Gaussians
- Challenge 2: Insufficient excitement of the latent model system for the first ℓ several steps
  - Linear regression with covariates whose covariances are rank-deficient, and with correlated noise
  - Latent model can only be partially identified in certain directions (but was proven to be enough)
- Challenge 3: Matrix factorization — need a new Procrustes-type lemma (due to rank-deficiency)

$$\min_{M_t, A_t, B_t, b_t} \sum_{i=1}^{n} \underbrace{\left(\|M_t h_t^{(i)}\|^2 + \sum_{\tau=t}^{t+m-1} \|u_\tau^{(i)}\|_{R_t}^2 + b_t - \sum_{\tau=t}^{t+m-1} c_t^{(i)} \right)^2}_{\textbf{cost prediction error}} + \cancel{\underbrace{(M_{t+1} h_{t+1}^{(i)} - A_t M_t h_t^{(i)} - B_t u_t^{(i)})^2}_{\textbf{transition prediction error}}}$$

Extension: MuZero-style for LTI systems

- MuZero (Schrittwieser et al., 2020) supersedes AlphaGo (Silver et al., 2016), AlphaGo Zero (Silver et al., 2017) and AlphaZero (Silver et al., 2018), as a "general game player" —
  - Matches the superhuman performance of AlphaZero in Go, shogi and chess, while outperforming model-free RL algorithms in Atari games
  - Key algorithmic components: Latent Model Learning + Monte-Carlo Tree Search
  - Viewed as a milestone of representation learning for control in deep RL
- Our latent model learning is not exactly the same as that in MuZero
  - Ours (explicit) — solve least-squares on latent states: $(\hat{A}, \hat{B}) \in \arg\min_{A,B} \sum_i \|Az_t^{(i)} + Bu_t^{(i)} - z_{t+1}^{(i)}\|^2$
  - MuZero-style (implicit) — by predicting the "cost" at future states, i.e., also "cost-driven"

$$\min_{M,A,B,b} \sum_{t=H}^{T+H-1} \left((\|Mh_t\|^2 + b - c_t)^2 + (\|AMh_t + Bu_t\|^2 + b - c_{t+1})^2 \right)$$

- This approach also works for LTI LQG control (when predicting "cumulative cost", as before)!

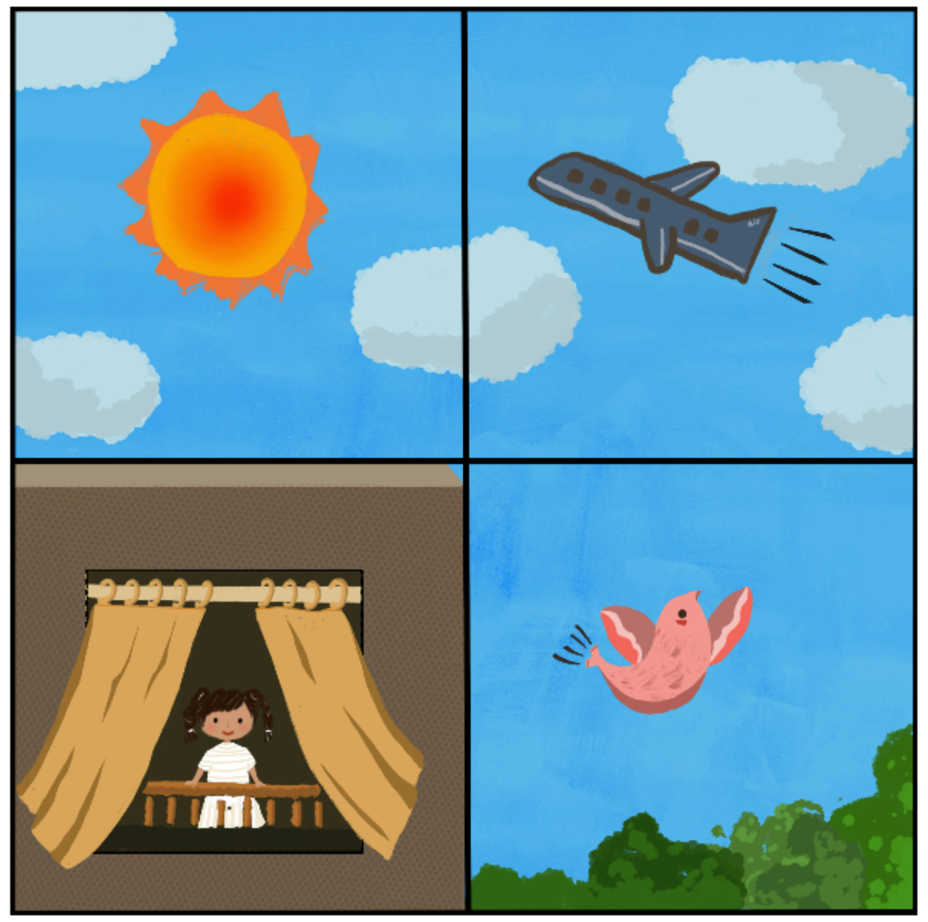
A Few More Highlights

- What makes a good "(Information) State" — sufficient statistics for optimal decision-making — has always existed in Stochastic Control literature (Striebel, 1965; Kwakernaak, 1965; Witsenhausen, 1976; Kumar and Varaiya, 1986; Mahajan, 2008; Adlakha, Lall, Goldsmith, 2012)
- Approximate information state (AIS): (Subramanian et al., 2022)
  - What is the right (sufficient) conditions for an "approximate sufficient statistics"
  - It has to predict both "(single-step) reward" and "itself" well
- State-based v.s. History-based representation learning: (Ni et al., 2024)
  - Bridging the desiderata and languages from empirical (deep) RL and Control
  - Key idea: "self-prediction"
- Cost-driven/MuZero-Style methods beyond linear quadratic case?
  - It doesn't work for discrete-space case! (Jiang, 2024)
- It is not a completely new idea! We had "Identification for Control" (I4C) (Gevers, 2005)

A Few More Highlights: Ongoing — A Unified Theory

Objective	Learned state space (well-defined in Controls literature)
Observation-driven	Full state space
Cost-driven	Cost observable subspace
Action-driven	Controllable subspace

task-relevant
task-irrelevant
uncontrollable
controllable



Goal: Let more sunlight in

- Consolidate the intuition: different objectives work differently, with pros and cons
  - Observation-driven: retains the most, but suffer from "noisy TV" issue (control-irrelevant information)
  - Cost-driven: minimum subspace for optimal control, but may not generalize across tasks
  - Action-driven: controllable subspace, but may not be enough for optimal control (e.g., when cost only cares uncontrollable subspace)
  - Can be viewed as approaches to (partial) system identification for control (I4C)

Source: "Denoised MDPs: Learning World Models Better Than the World Itself".

Concluding Remarks

Concluding Remarks

- What is a good state(-space) for practical control/RL tasks? What is the right objective to learn it?
- The state representation of LQG can be learned by predicting (cumulative) costs
  - Insights into Value Prediction Network (Oh et al., 2017) & MuZero (Schrittwieser et al., 2020)

Concluding Remarks

- What is a good state(-space) for practical control/RL tasks? What is the right objective to learn it?
- The state representation of LQG can be learned by predicting (cumulative) costs
  - Insights into Value Prediction Network (Oh et al., 2017) & MuZero (Schrittwieser et al., 2020)
  - Many open questions in State-Representation Learning for Control — requires bridging ideas and insights from both Control and Learning

Thanks!