Reinforcement Learning: States Representation, Policy Convergence, and Robustness.

Lekan Molu

Presented by **Lekan Molu** (Lay-con Mo-lu)

September 2, 2025

Technical Overview

This page is left blank intentionally.

Controllable States Retrieval in RL

An Agentic Collision-Free Mapping System

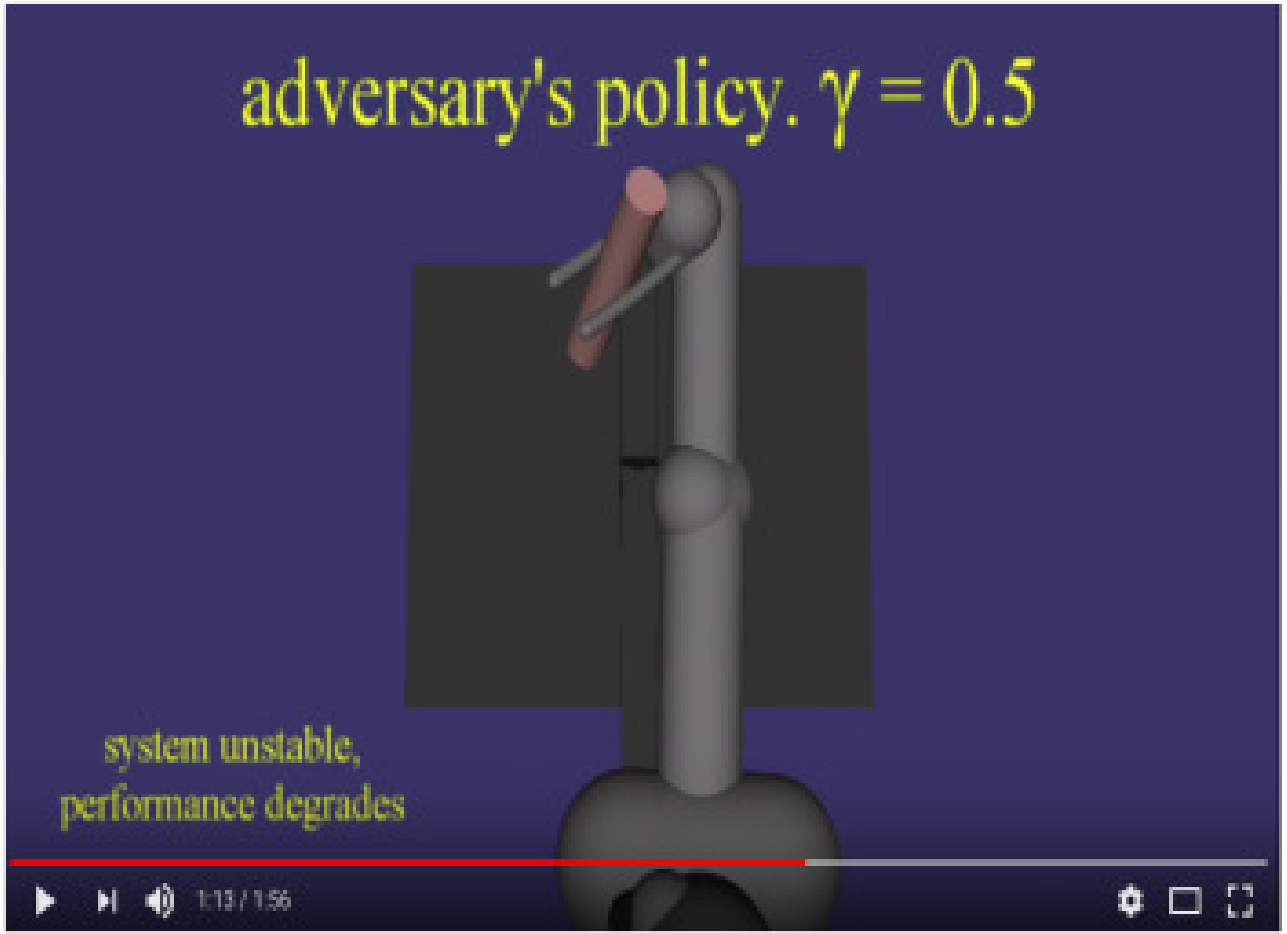
A Fast Universal Collision-free Agentic Model: Compact Illusory Representation and Memory-Efficient Incremental Mapping.

***Abstract*— We present a generalist collision-free computational agent that rapidly embeds acquired knowledge about real world environments into its internal model or "head" based on a pipeline of machine learning ensembles wrapped around a sequence of GPU-accelerated approximate convex decomposition, a probabilistic convex set polytopes computational scheme, frontier-based planning schemes, and low-level nonlinear control for general computational geometry navigation tasks. The agent's head stores a compact, memory-efficient, and computationally tractable internal model of the environment that *proactively* constructs a collision-free model based on exogenous perception, updates and maintains acquired state, whilst adaptively modifying erstwhile computed states based on newly retrieved information from external stimuli. This agentic design offers flexibility in many real-time applications and encourages self-collision awareness, rapid fault diagnosis and recovery in complex environments, and provides an efficient storage mechanism that makes it suited to long-range mapping and tracking systems. We train the agent on diverse dataset and leverage machine learning in reasoning through an ensemble of established and novel computational geometry, control systems, and convex optimization algorithms. Such a model is increasingly becoming a necessity in many 'embodied agentic' and emergent real-time AI applications. This work serves as a first step in answering the call for a foundational generalist agent that uses all perception information available for its autonomous exploration (vision, tactile, ranging etc) in rapidly building a *correct and minimal* environment representation that can then be leveraged for further downstream tasks**

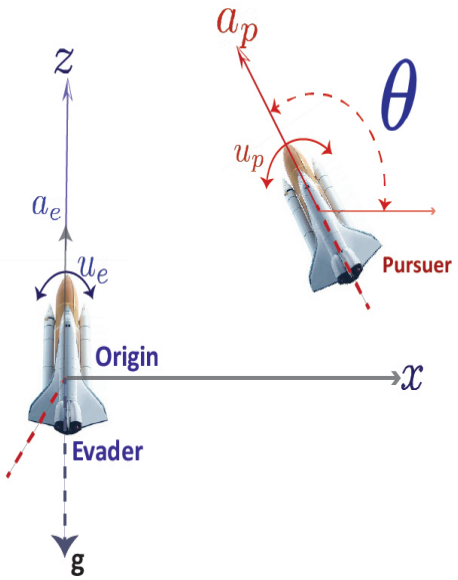
than its specific model instances that may lack completeness or correctness. The internal model principle in neuroscience [16] is a splendid inspiration in our work for modeling autonomous reasoning agents with *compact, memory-efficient internal information on the external world*. In this internal model principle, if an agent possesses a *minimum model* of "external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it" [16]. With Craik [16]'s view that the nervous system is a calculating machine with a modeling or paralleling capacity for external events that engenders thought and of explanation [16], we seek to build agents with the representation power above that generalize into the diverse tasks earlier enumerated.

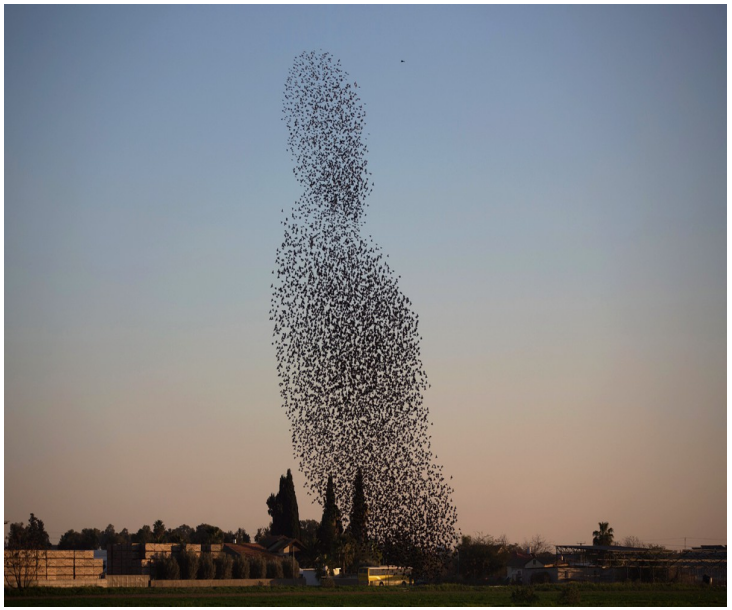
Large deep models and policies have emerged as viable mechanisms for encoding such perceptual experiences and building these agentic systems [88]. Two application categories have broadly emerged in these large agentic paradigms: (1) situated agents i.e. agents trained on stored information in the form of static data albeit without an

Policies' Brittleness Quantification in RL



Numerical safety analysis in dynamical Systems





Numerical safety analysis in dynamical Systems

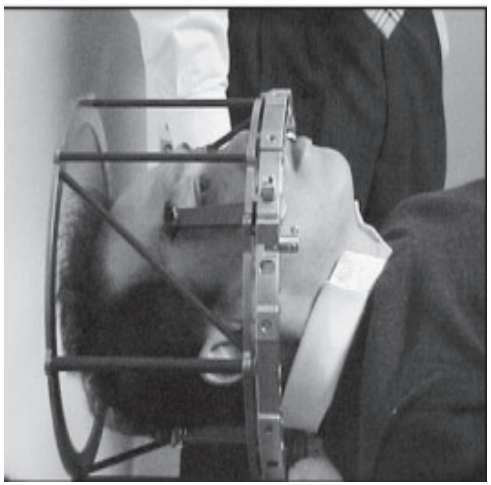
LevelSetPy: A GPU-Accelerated Package for Hyperbolic Hamilton-Jacobi Partial Differential Equations

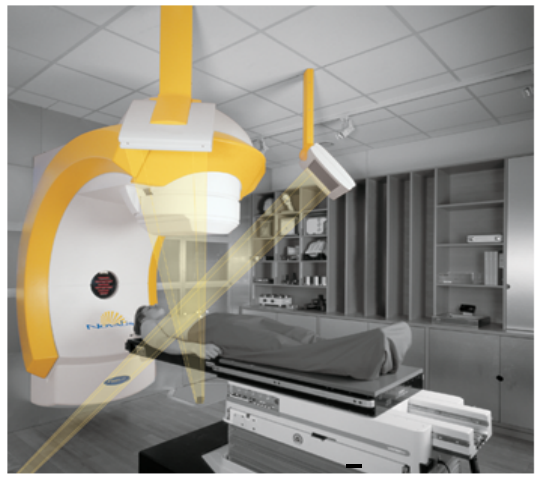
LEKAN MOLU, Microsoft Research, USA

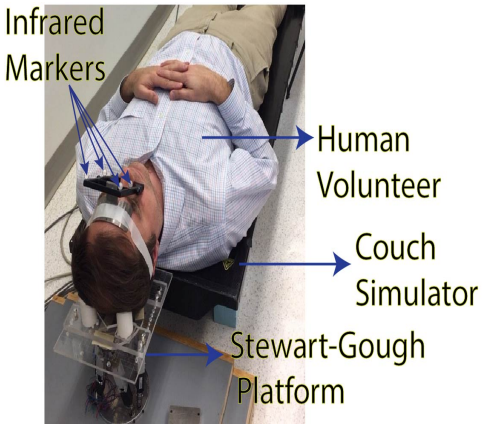
This article introduces a software package release for geometrically reasoning about the *safety* desiderata of (complex) dynamical systems via level set methods. In emphasis, safety is analyzed with Hamilton-Jacobi equations. In scope, we provide implementations of numerical algorithms for the resolution of Hamilton-Jacobi-Isaacs equations: the spatial derivatives of the associated value function via upwinding, the Hamiltonian via Lax-Friedrichs schemes, and the integration of the Hamilton-Jacobi equation altogether via total variation diminishing Runge-Kutta schemes. Since computational speed and interoperability with other modern scientific computing libraries (typically written in the Python language) is of essence, we capitalize on modern computational frameworks such as `CUPY` and `NUMPY` and move heavy computations to GPU devices to aid parallelization and improve bring-up time in safety analysis. We hope that this package can aid users to quickly iterate on ideas and evaluate all possible safety desiderata of a system via geometrical simulation in modern engineering problems.

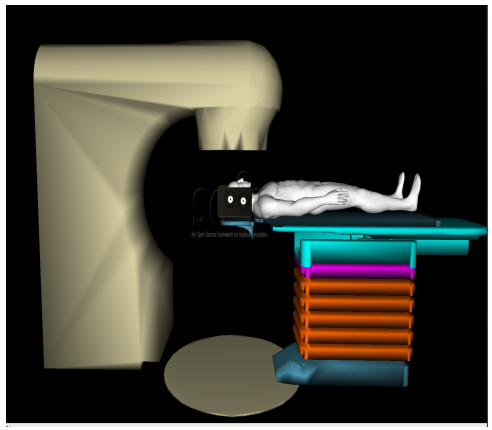
CCS Concepts: • **Software and its engineering** → **Software libraries and repositories**; • **Applied computing** → *Physical sciences and engineering*; • **Mathematics of computing** → *Solvers*.

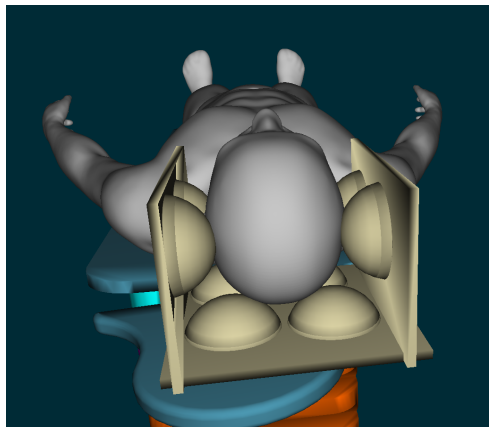
Patient Head Stabilization in IGRT











State Representation in RL: Credits

S. Chen

A. Koul

Y. Efroni

D. Misra

D. Foster

R. Islam

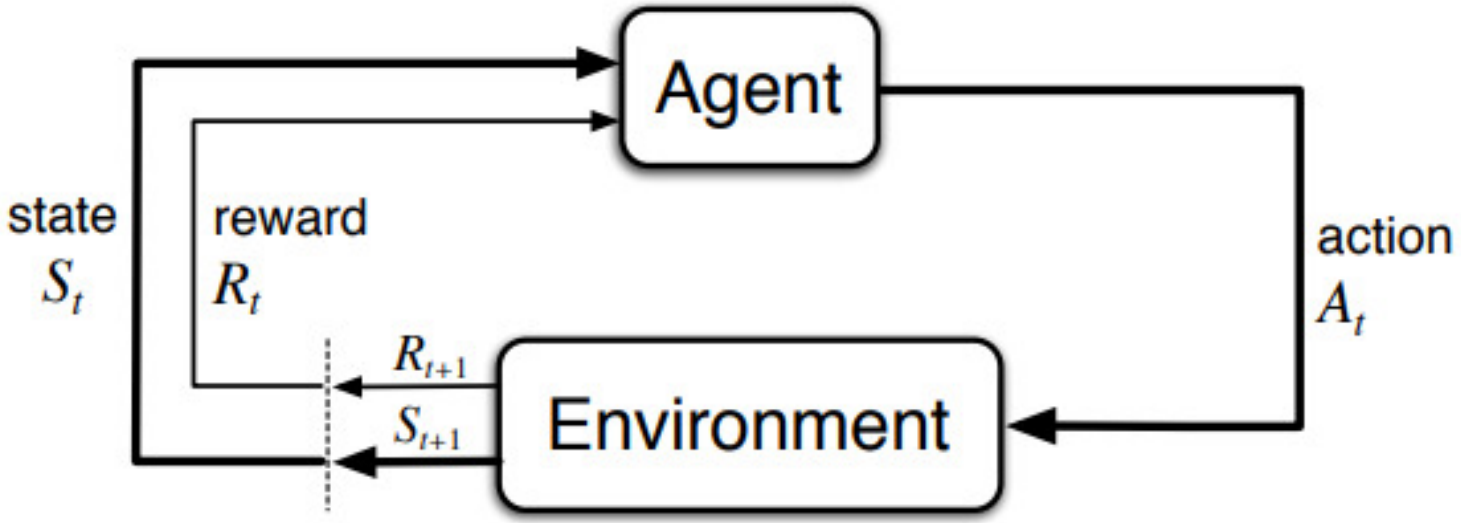
A. Lamb

M. Dudik

A. Krish.

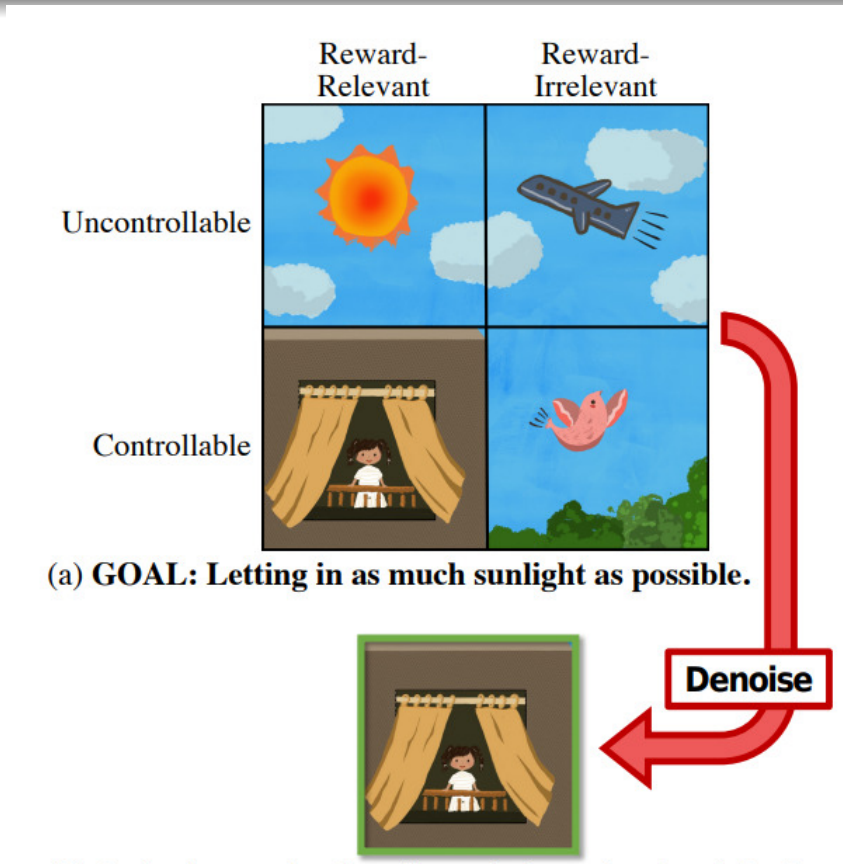
J. Langford

Standard Reinforcement Learning



3

Compact States without Exogenous Distractors

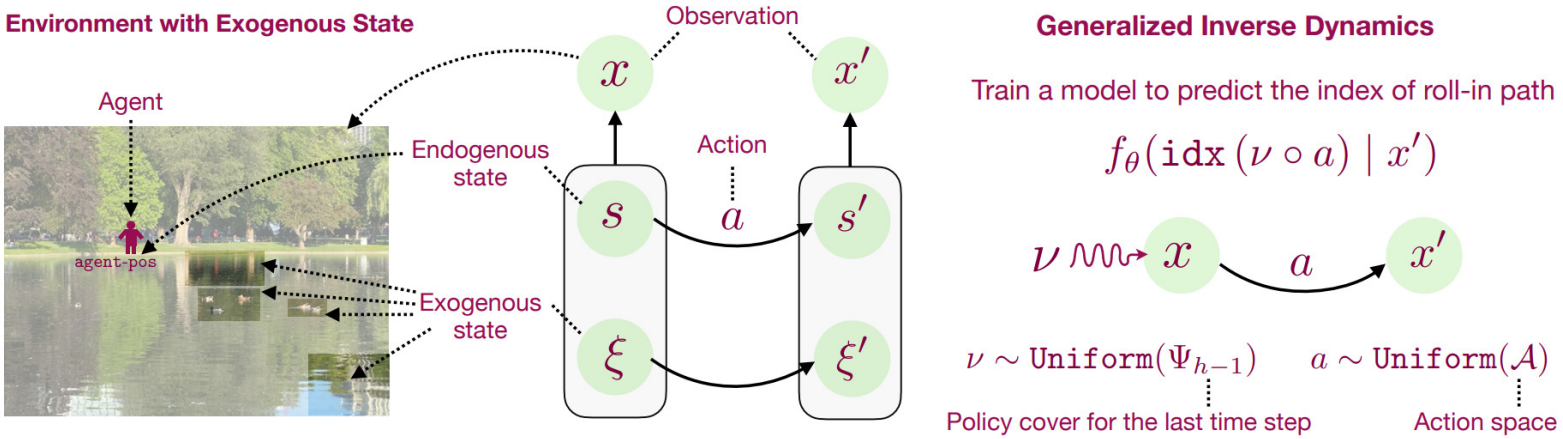


(a) **GOAL: Letting in as much sunlight as possible.**

(b) Optimal control only relies on information that is **both controllable and reward-relevant**. Good world models should ignore other factors as noisy distractors.

Denoised MDPs: Learning World Models Better Than the World Itself [5]

Compact States without Exogenous Distractors



Learning s with $[S]$ whilst ignoring temporally correlated ξ? Source: [3, Fig. 1].

Exo-MDP Machinery

- Consider the tuple $\mathcal{M} := (\mathcal{X}, \mathcal{Z}, \mathcal{A}, T, R, H)$
  - Starting distribution $\mu \in \Delta(\mathcal{Z})$;
  - Agent receives observations $\{x_h\}_{h=1}^{H} \in \mathcal{X}$ from the emission function $q : \mathcal{Z} \to \Delta(\mathcal{X})$;
  - Agent transitions between latent states via $T : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{S})$;
    - And rewards by $R : \mathcal{X} \times \mathcal{A} \to \Delta([0, 1])$
- Trajectories: $(z_1, x_1, a_1, r_1, \cdots, z_H, a_H, r_H)$ from repeated interactions;
  - $z_1 \sim \mu_1(\cdot)$, $z_{h+1} \sim T(\cdot|z_h, a_h)$, $x_h \sim q(\cdot|z_h)$ and $r_h \sim R(x_h, a_h, x_{h+1})$ for all $h \in [H]$.
- Define $supp(q(\cdot|z)) = \{x \in \mathcal{X} | q(x|z) > 0\}$ for any z.

Exo-MDP Machinery

Block MDP assumption $supp(q(\cdot|z_1)) \cap sup(q(\cdot|z_2)) = \emptyset$ for all $z_1 \neq z_2$.

- Agent chooses $a \sim \pi(z_h|x_h)$
- There exists non-stationary episodic policies $\Pi_{NS} := \Pi^H \supseteq (\pi_1, \cdots, \pi_H)$;
- Optimal policy $\pi^\star = \text{argmax} V_{\pi \in \Pi_{NS}}(\pi)$;
  - For $V_{\pi \in \Pi_{NS}} = \sum_h = 1^H r_h$.
- EXO-BMDP: Essentially a Block MDP [1] such that the latent states admits the form $z = (s, e)$, where $s \in \mathcal{S}$, $e \in E$.
- $\mu(z) = \mu(s)\mu\xi$ and $T(z'|z, a) = T(s'|s, a)\, T_e(e'|e)$

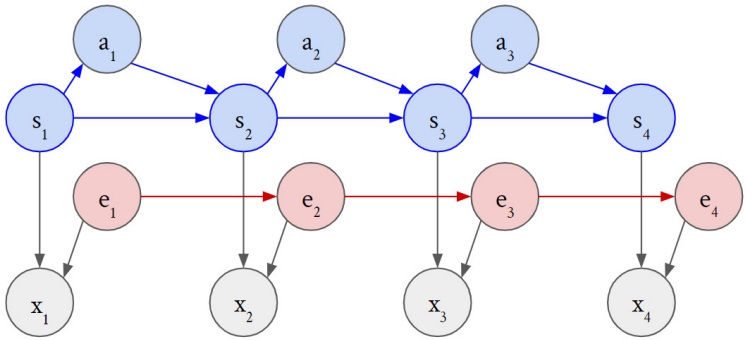
Literature comparison

Algorithms	PPE	OSSR	DBC	CDL	Denoised-MDP	1-Step Inverse	AC-State (Ours)
Exogenous Invariant State	✓	✓	✓	✓	✓	✓	✓
Exogenous Invariant Learning	✓	✓	✗	✗	✗	✓	✓
Flexible Encoder	✓	✗	✓	✗	✓	✓	✓
YOLO (No Resets) Setting	✗	✓	✓	✓	✓	✓	✓
Reward Free	✓	✓	✗	✓	✓	✓	✓
Control-Endogenous Rep.	✓	✓	✗	✓	✓	✗	✓

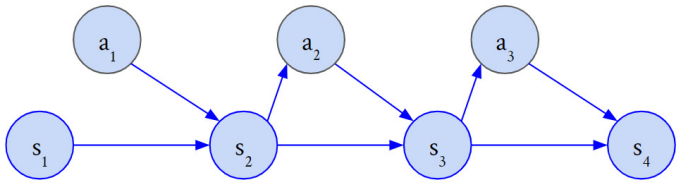
Emphasis on robustness to exogenous information. Comparison with baselines including PPE [3], OSSR [2], DBC [6] , Denoised MDP [5] and One-Step Inverse Models [4].

Rewards-agnostic Exogenous State Invariance in RL

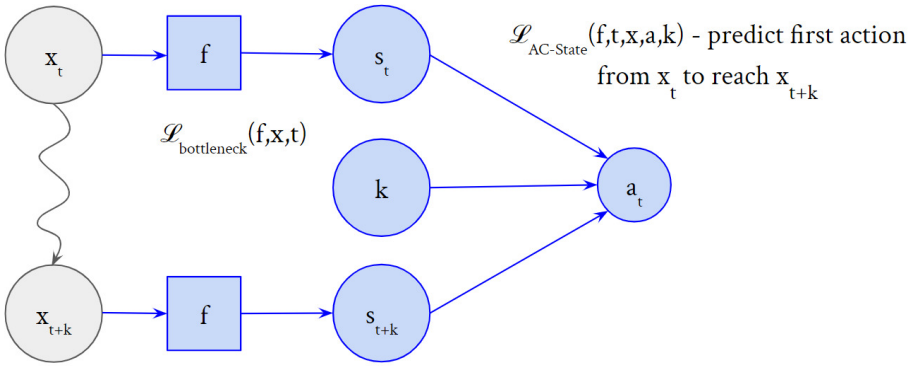
AC-State Discovers the smallest control-endogenous state s assuming factorized dynamics



AC-State collects data with a single random action followed by a high-coverage endogenous policy for k-1 steps



AC-State learns an encoder f for s = f(x) by optimizing a multi-step inverse model with a bottleneck



Latent States Discovery – Multi-step Inverse Dynamics

- $\hat{f} \approx \arg\min_{f \in \mathcal{F}} \mathbb{E}_{t,k} \Big[\mathcal{L}_{\mathrm{ACS}}(f, x, a, t, k) + \mathcal{L}_{\mathrm{B}}(f, x_t) + \mathcal{L}_{\mathrm{B}}(f, x_{t+k}) \Big]$

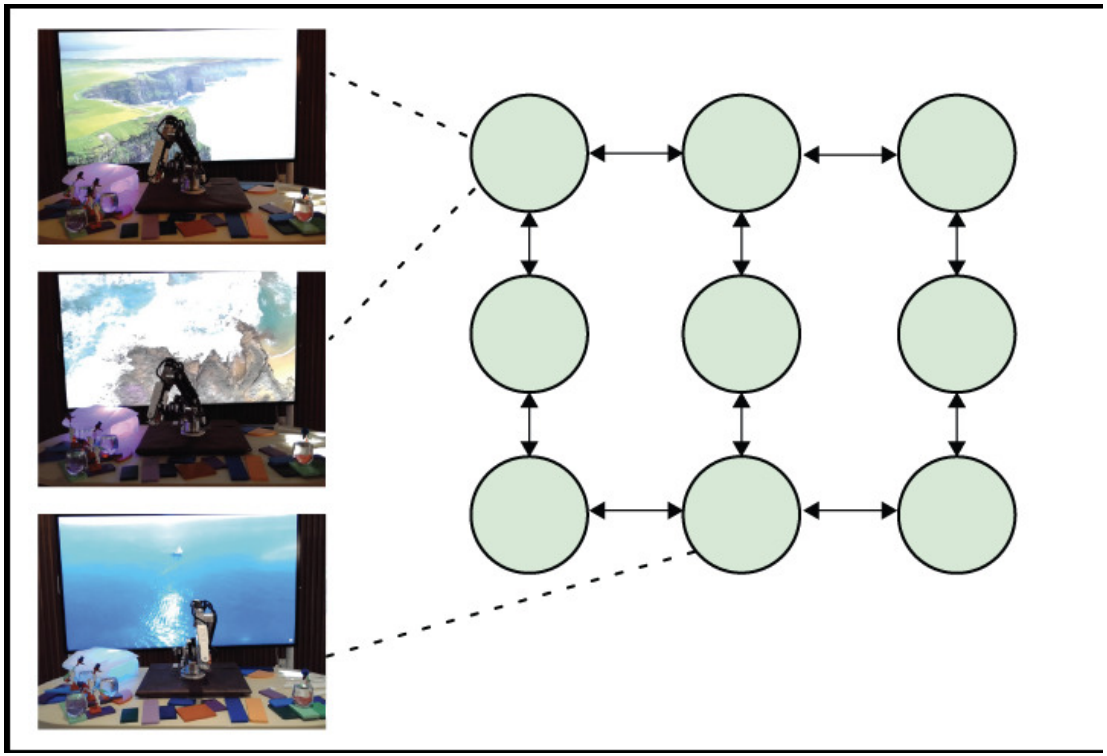
$$\mathcal{L}_{\mathrm{ACS}}(f, x, a, t; k) = -\log(\mathbb{P}(a_t | f(x_t), f(x_{t+k}); k)) \quad (1)$$

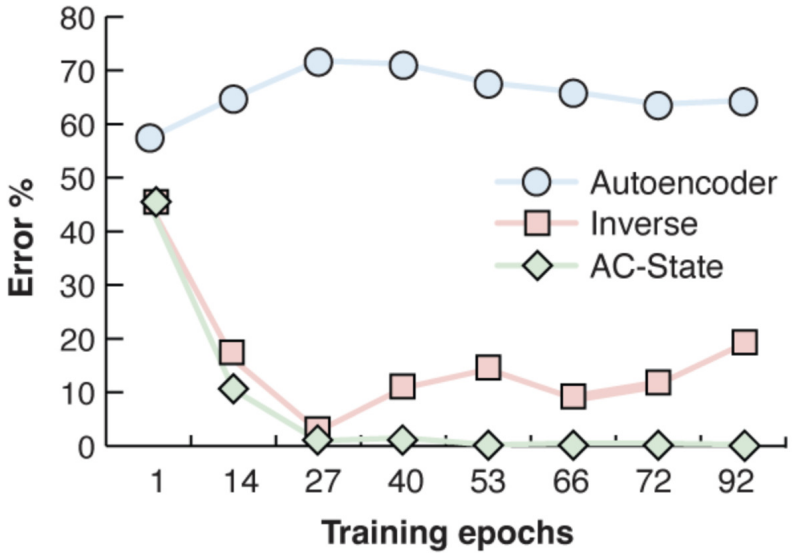
AC State Algorithm

Algorithm 1 AC-State Algorithm for Latent State Discovery Using a Uniform Random Policy

Initialize observation trajectory x and action trajectory a. Initialize encoder f_θ. Assume any pair of states are reachable within exactly K steps and a number of samples to collect T, and a set of actions $\mathcal{A}$, and a number of training iterations N.
$x_1 \sim U(\mu(x))$
for t = 1, 2, ..., T **do**
    $a_t \sim U(\mathcal{A})$
    $x_{t+1} \sim \mathbb{P}(x'|x_t, a_t)$
for n = 1, 2, ..., N **do**
    $t \sim U(1, T)$ and $k \sim U(1, K)$
    $\mathcal{L} = \mathcal{L}_{\text{AC-State}}(f_\theta, t, x, a, k) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_t) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_{t+k})$
    Update θ to minimize $\mathcal{L}$ by gradient descent.

AC State in Action

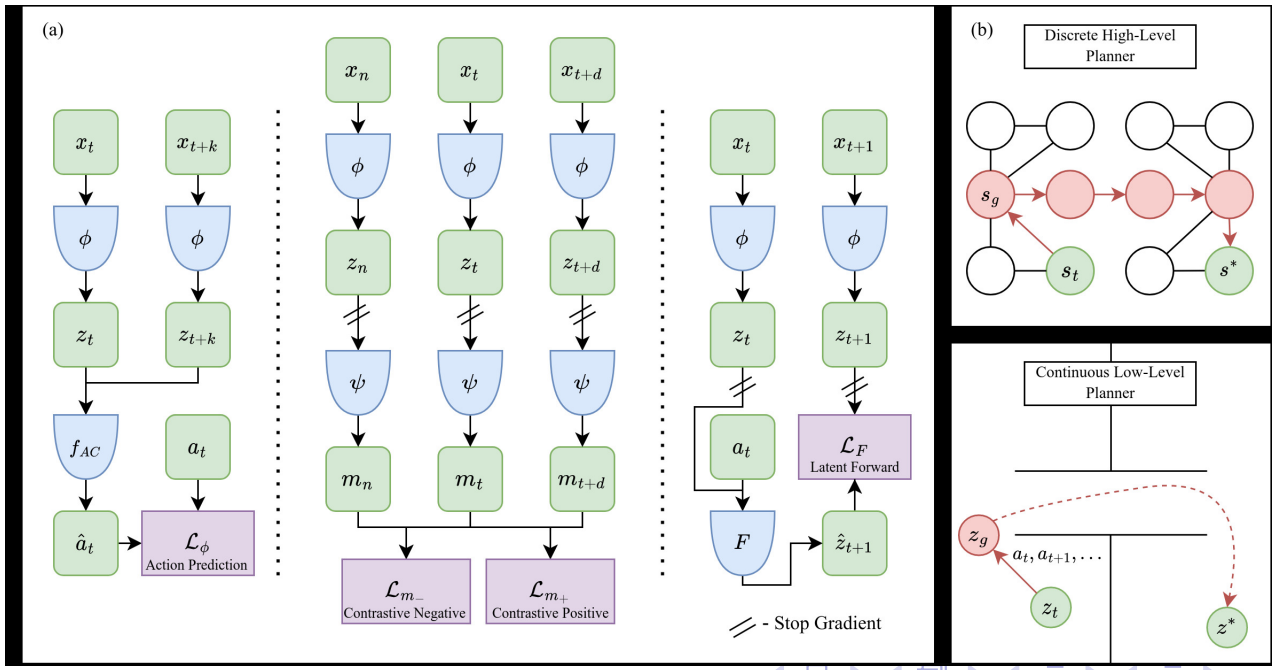




PCLAST: Agent Plannable Continuous Latent States

PcLast: Discovering Plannable Continuous Latent States

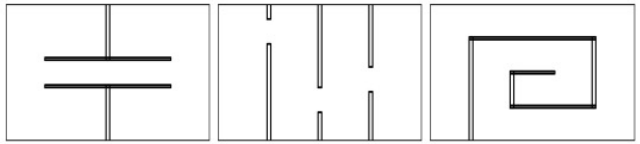
Anurag Koul [*1] Shivakanth Sujit [*2 3 4] Shaoru Chen [1] Ben Evans [5] Lili Wu [1] Byron Xu [1] Rajan Chari [1]
Riashat Islam [3 6] Raihan Seraj [3 6] Yonathan Efroni [7] Lekan Molu [1] Miro Dudik [1] John Langford [1] Alex Lamb [1]



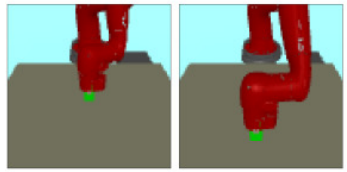
PCLAST Algorithm

Algorithm 1 n-Level Planner

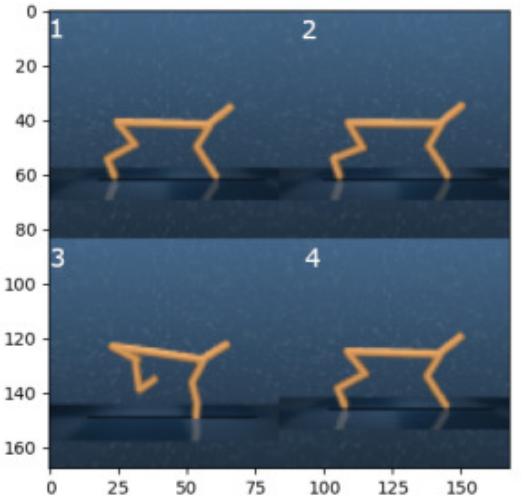
Require:
Current observation x_t
Goal observation x_{goal}
Planning horizon H
Encoder $\phi(\cdot)$
PCLAST map $\psi(\cdot)$
Latent forward dynamics $\delta(\cdot, \cdot)$
Multi-Level discrete transition graphs $\{\mathcal{G}_i\}_{i=2}^{n}$
Ensure: Action sequence $\{a_i\}_{i=0}^{H-1}$
1: Compute current continuous latent state $\hat{s}_t = \phi(x_t)$ and target latent state $\hat{s}^* = \phi(x_{goal})$.
{See Appendix E for details of high-level planner and low-level planner.}
2: **for** $i = n, n-1, \ldots, 2$ **do**
3: $\hat{s}^* =$ high-level planner$(\hat{s}_t, \hat{s}^*, \mathcal{G}_i)$
{Update waypoint using a hierarchy of abstraction.}
4: **end for**
5: $\{a_i\}_{i=0}^{H-1} =$ low-level planner$(\hat{s}_t, \hat{s}^*, H, \delta, \psi)$
{Solve the trajectory optimization problem.}



(a) Hallway (b) Rooms (c) Spiral



(d) Sawyer Reach Environment



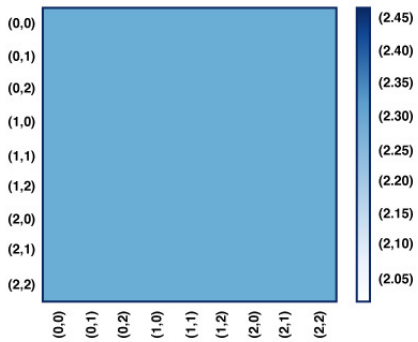
PCLAST Results

Method	Reward type	Hallway	Rooms	Spiral	Sawyer-Reach
PPO	Dense	6.7 ± 0.6	7.5 ± 7.1	11.2 ± 7.7	$\mathbf{86.00 \pm 5.367}$
PPO + ACRO	Dense	10.0 ± 4.1	23.3 ± 9.4	23.3 ± 11.8	84.00 ± 6.066
PPO + PCLaSt	Dense	$\mathbf{66.7 \pm 18.9}$	$\mathbf{43.3 \pm 19.3}$	$\mathbf{61.7 \pm 6.2}$	78.00 ± 3.347
PPO	Sparse	1.7 ± 2.4	0.0 ± 0.0	0.0 ± 0.0	68.00 ± 8.198
PPO + ACRO	Sparse	21.7 ± 8.5	5.0 ± 4.1	11.7 ± 8.5	$\mathbf{92.00 \pm 4.382}$
PPO + PCLaSt	Sparse	$\mathbf{50.0 \pm 18.7}$	$\mathbf{6.7 \pm 6.2}$	$\mathbf{46.7 \pm 26.2}$	82.00 ± 5.933
CQL	Sparse	3.3 ± 4.7	0.0 ± 0.0	0.0 ± 0.0	32.00 ± 5.93
CQL + ACRO	Sparse	15.0 ± 7.1	$\mathbf{33.3 \pm 12.5}$	$\mathbf{21.7 \pm 10.3}$	68.00 ± 5.22
CQL + PCLaSt	Sparse	$\mathbf{40.0 \pm 0.5}$	23.3 ± 12.5	20.0 ± 8.2	$\mathbf{74.00 \pm 4.56}$
RIG	None	0.0 ± 0.0	0.0 ± 0.0	3.0 ± 0.2	$\mathbf{100.0 \pm 0.0}$
RIG + ACRO	None	$\mathbf{15.0 \pm 3.5}$	$4.0 \pm 1.$	$\mathbf{12.0 \pm 0.2}$	100.0 ± 0.0
RIG + PCLaSt	None	10.0 ± 0.5	4.0 ± 1.8	10.0 ± 0.1	90.0 ± 5
Low-Level Planner + PCLaSt	None	86.7 ± 3.4	69.3 ± 3.4	50.0 ± 4.3	$\pm$
n-Level Planner + PCLaSt	None	$\mathbf{97.78 \pm 4.91}$	$\mathbf{89.52 \pm 10.21}$	$\mathbf{89.11 \pm 10.38}$	95.0 ± 1.54

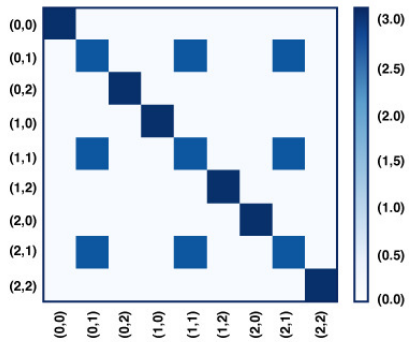
AC State in Action

Exogenous distractors riddance.

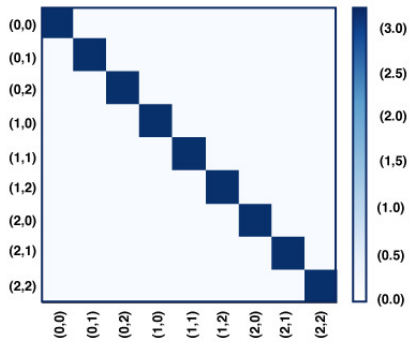
Agent Controllable States Representation



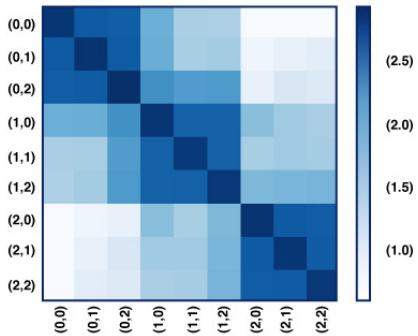
(a) Autoencoder (Theory worst-case)



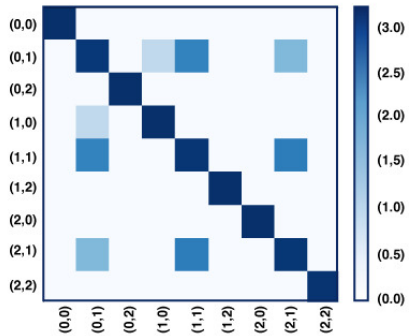
(b) Inverse (Theory worst-case)



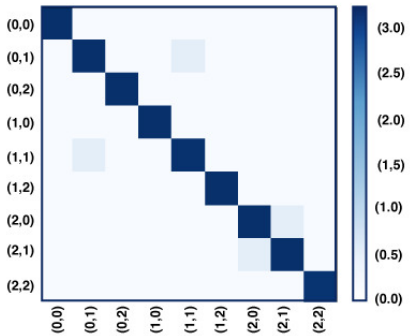
(c) AC-State (Theory worst-case)



(d) Autoencoder (Empirical)

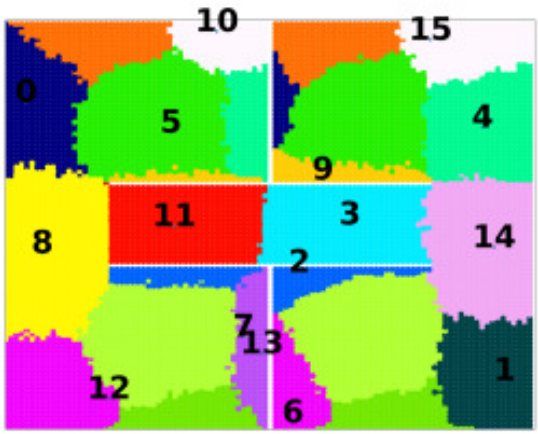


(e) Inverse (Empirical)



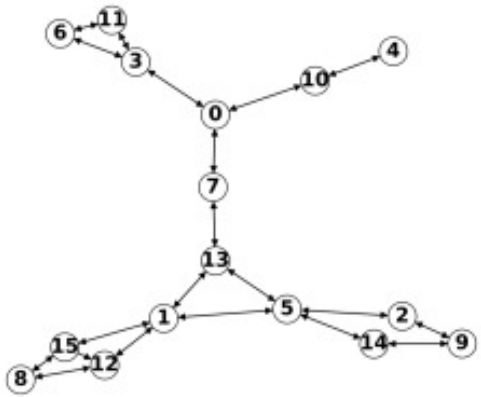
(f) AC-State (Empirical)

PCLAST Segmentation Results

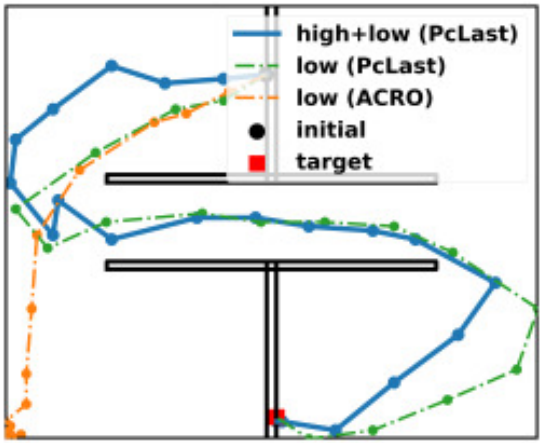


(a) Clusters ACRO

(b) Clusters PCLAST

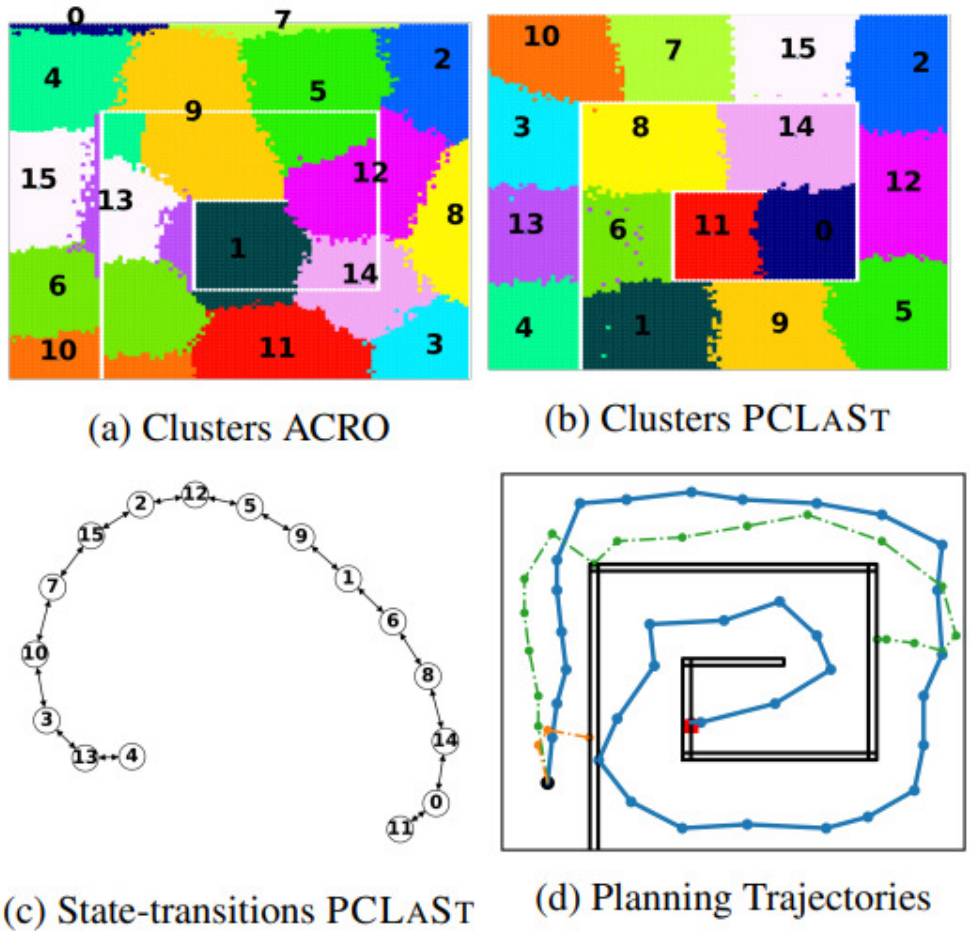


(c) State-transitions PCLAST



(d) Planning Trajectories

PCLAST Segmentation Results



(a) Clusters ACRO

(b) Clusters PCLAST

(c) State-transitions PCLAST

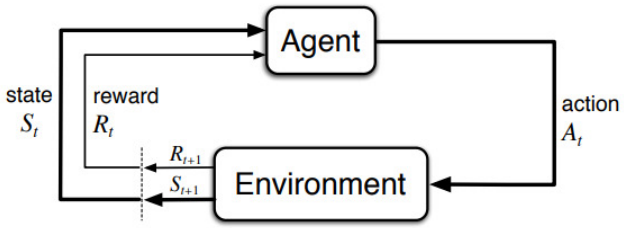
(d) Planning Trajectories

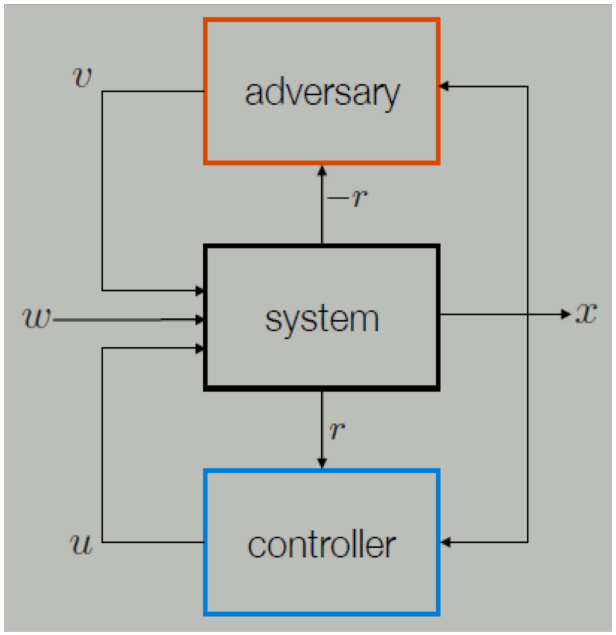
Figure 6. Clustering, Abstract-MDP, and Planning are shown for

Iterative Dynamic Game in RL

This page is left blank intentionally.

Inculcating robustness into multistage decision policies





Problem Setup

- To quantify the brittleness, we optimize the stage cost

$$\max_{\boldsymbol{v}_t \sim \psi \in \Psi} \left[\sum_{t=0}^{T} \underbrace{c(\boldsymbol{x}_t, \boldsymbol{u}_t)}_{\text{nominal}} - \gamma \underbrace{g(\boldsymbol{v}_t)}_{\text{adversarial}} \right]$$

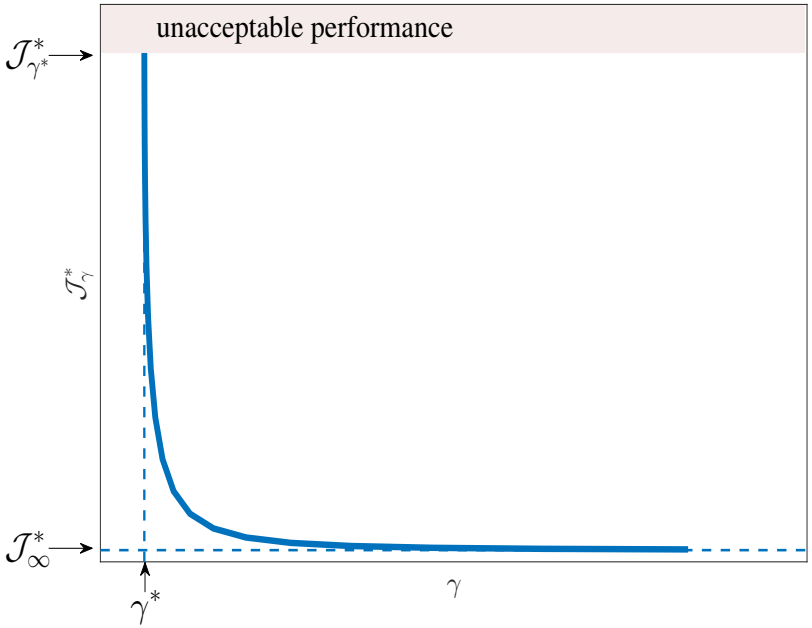
- To mitigate lack of robustness, we optimize the *cost-to-go*

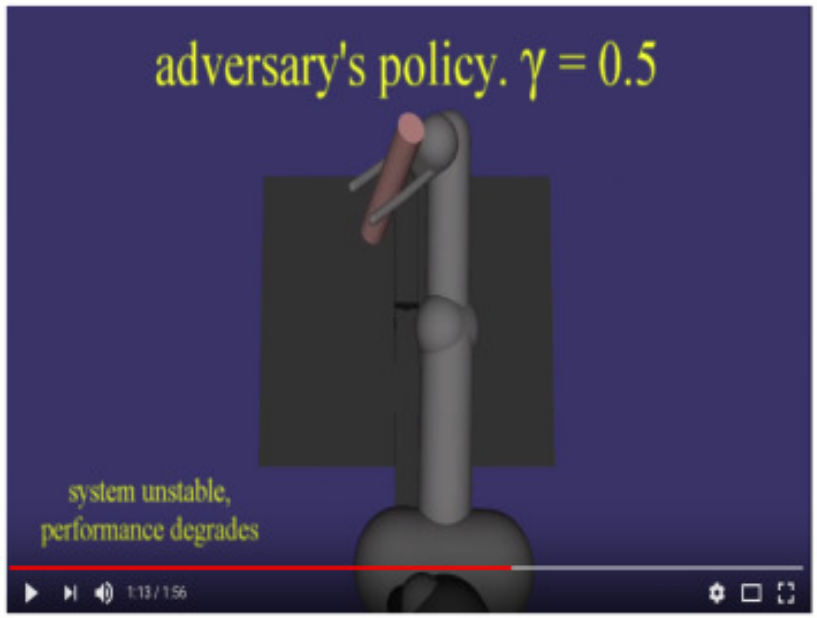
$$c_t(\boldsymbol{x}_t, \pi, \psi) = \min_{\boldsymbol{u}_t \sim \pi} \max_{\boldsymbol{v}_t \sim \psi} \left(\sum_{t=0}^{T-1} \ell_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{v}_t) + L_T(\boldsymbol{x}_T) \right),$$

- and seek a saddle point equilibrium policy that satisfies

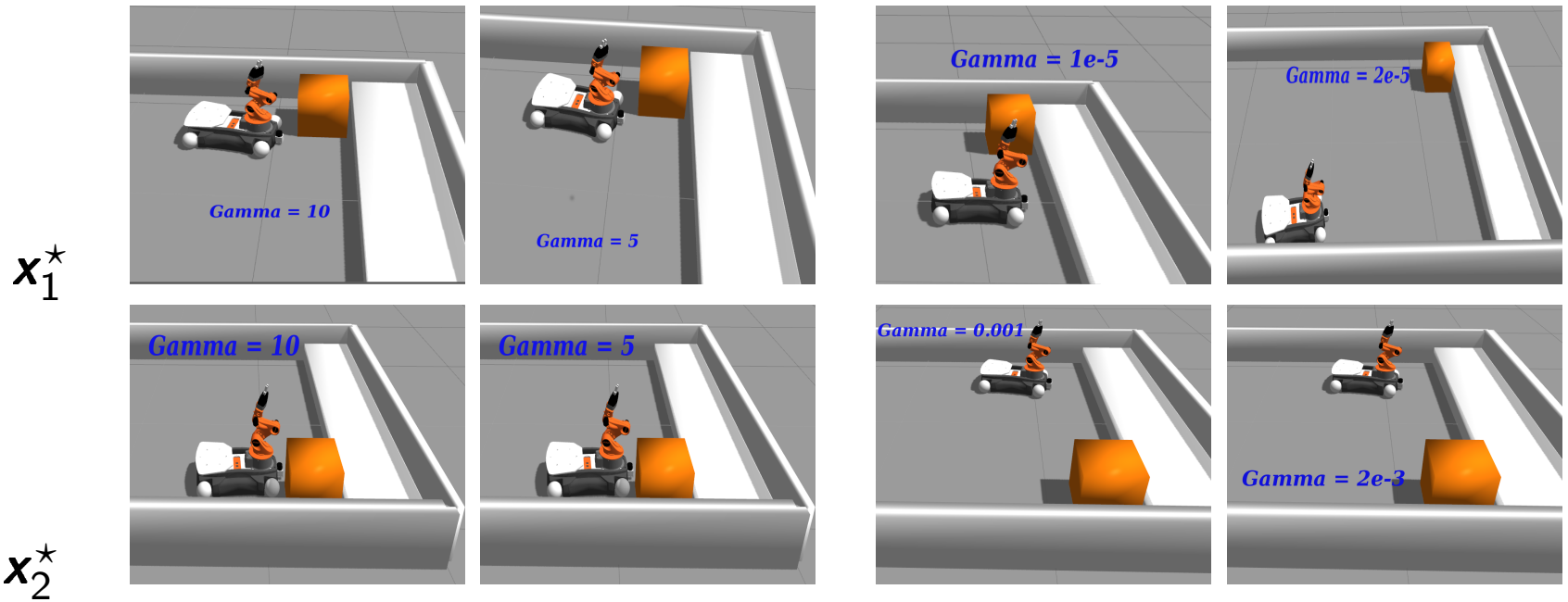
$$c_t(\boldsymbol{x}_t, \pi^*, \psi) \leq c_t(\boldsymbol{x}_t, \pi^*, \psi^*) \leq c_t(\boldsymbol{x}_t, \pi, \psi^*),$$

Results: Brittleness Quantification





Results: Iterative Dynamic Game



End pose of the KUKA platform with our iDG formulation given different goal states and γ-values.

Mixed H_2/H_∞ Policy Optimization in RL

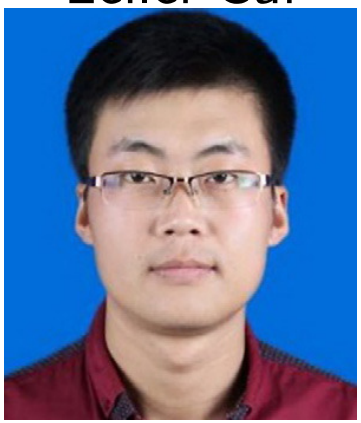
"The scientist's problem is to recognize basic facts even though they are obscured by a wealth of extraneous material, and then to apply creative imagination in their interpretation. This Karl Jansky did." – Cyril Jansky.

Talk Outline and Overview

- Policy Optimization and Stochastic Linear Control
  - Connections to risk-sensitive control;
  - Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control theory.
- The case for convergence analysis in stochastic PO.
  - Kleinman's algorithm, *redux.*
  - Kleiman's algorithm in an iterative best response setting;
  - PO Convergence in best response settings.
- Robustness margins in model- and sampling- settings.
  - PO as a discrete-time nonlinear system;
  - Kleiman and input-to-state-stability;
  - Robust policy optimization as a small-input stable state optimization algorithm

Credits

Leilei Cui



Postdoc, MIT

Zhong-Ping Jiang

Professor, NYU

Research Significance

- (Deep) RL and modern AI
  - Robotic manipulation (Levine et al., 2016), text-to-visual processing (DALL-E), Atari games (Mnih et al., 2013), e.t.c.
  - Policy optimization (PO) is fundamental to modern AI algorithms' success.
  - Major success story: functional mapping of observations to policies.
  - But how does it work?

Policy Optimization – Open questions

- Gradient-based data-driven methods: prone to divergence from true system gradients.
  - Challenge I: Optimization occurs in non-convex objective landscapes.
    - Get performance certificates as a mainstay for control design: Coerciveness property (Hu et al., 2023).
  - Challenge II: Taming PG's characteristic high-variance gradient estimates (REINFORCE, NPG, Zeroth-order approx.).
    - Hello, (linear) robust ($\mathcal{H}_\infty$-synthesis) control!

Policy Optimization – Open questions

- Challenge III: Under what circumstances do we have convergence to a desired equilibrium in RL settings?
- Challenge IV: Stochastic control, not deterministic control settings.
  - models involving round-off error computations in floating point arithmetic calculations; the stock market; protein kinetics.
- Challenge V: Continuous-time RL control.
  - Very little theory. Lots of potential applications encompassing rigid and soft robotics, aerospace or finance engineering, protein kinetics.

Tools: Complexity, Convergence, Robustness.

- Risk-sensitive $\mathcal{H}_\infty$-control (Glover, 1989) and discrete- and continuous-time mixed $\mathcal{H}_2/\mathcal{H}_\infty$ design (Khargonekar et al., 1988; Hu et al., 2023):
  - min. upper bound on $\mathcal{H}_2$ cost subject to satisfying a set of risk-sensitive (often $\mathcal{H}_\infty$) constraints (Basar, 1990):

$$\begin{aligned} &min_{K\in\mathcal{K}} J(K) := Tr(P_K DD^\top) \\ &\text{subject to } \mathcal{K} := \{K | \rho(A - BK) < 1, \|T_{zw}(K)\|_\infty < \gamma\} \end{aligned} \tag{2}$$

  - P_K: solution to the generalized algebraic Riccati equation (GARE);
  - A, B, D, K: standard closed-loop system matrices;
  - $\|T_{zw}(K)\|_\infty$: $\mathcal{H}_\infty$-norm of the closed-loop transfer function from a disturbance input w to output z.

Tools: Complexity, Convergence, Robustness.

Infinite-horizon

- discrete-time deterministic LQR settings (Fazel et al., 2018):

$$\min_{K\in\mathcal{K}} \mathbb{E}\sum_{t=0}^{\infty}(x_t^\top Q x_t + u_t^\top R u_t) \text{ s.t. } x_{t+1} = Ax_t + Bu_t, x_0 \sim \mathcal{P}_0$$

- discrete-time LQ problems under multiplicative noise (Gravell et al., 2021):
$\min_{\pi\in\Pi} \mathbb{E}_{x_0,\{\delta_i\},\{\gamma_i\}} \sum_{t=0}^{\infty}(x_t^\top Q x_t + u_t^\top R u_t)$
subject to $x_{t+1} = (A+\sum_{i=1}^{p}\delta_{ti}A_i)x_t + (B+\sum_{i=1}^{q}\gamma_{ti}B_i)u_t;$

(Non-exhaustive) Lit. Landscape on PO Theory

Literature landscape	Cont. time (Kalman '61, Luenberger '63)	Stochastic. LQR (Kalman '60)	Cont. Phase	LEQG or Mixed H_2/H_∞	Finite/Infinite Horizon
Fazel (2018)	No	No	Yes	No	Finite-horizon
Mohammadi (TAC -- 2020)	Yes	No	Yes	No	Finite-Horizon
Zhang (2019)	Yes	Yes (Gaussian)	Yes	Yes	Inf-horizon
Gravell (2021)	No	Multiplicative	Yes	No	Inf-horizon
Zhang (2020)	No	No	Yes	Yes	Rand-horizon
Molu (2022)	Yes	Yes (Brownian)	Yes	Yes	Inf-Horizon
Cui & Molu (2023)	Yes	Yes (Brownian)	Yes	Yes	Inf-Horizon

Mainstay

- Continuous-time infinite-dimensional linear systems.
  - Disturbances enter additively as random stochastic Wiener processes.
  - Many natural systems admit uncertain additive Brownian noise as diffusion processes.
    - Theoretical analysis machinery: Îto's stochastic calculus.
- Goal: keep controlled process, z, small i.e.

$$\|z\|_2 = \left(\int |z(t)|^2 dt \right)^{1/2},$$

  - Under a minimizing $u(x(t)) \in \mathcal{U}$ in spite of unforeseen $w(t) \in \mathcal{W} \subseteq \mathbb{R}^q$.

Minimization Objective and Risk-Sensitive Control

- Risk-sensitive linear exponential quadratic Gaussian objective functional (Jacobson, 1973):

$$\min_{u\in\mathcal{U}} \mathcal{J}_{exp}(x_0, u, w) = \mathbb{E}\Big|_{x_0\in\mathcal{P}_0} \exp\left[\frac{\alpha}{2}\int_0^\infty z^\top(t)z(t)\mathrm{d}t\right],$$
$$\text{subject to } \mathrm{d}x(t) = Ax(t)\mathrm{d}t + Bu(t)\mathrm{d}t + D\mathrm{d}w(t),$$
$$z(t) = Cx(t) + Eu(t),\ \alpha > 0; \tag{3}$$

- where $dw/dt = \mathcal{N}(0, W)$, $x_0 = \mathcal{N}(0, \mu)$, and $(x_0, w(t)) \subseteq (\Omega, \mathcal{F}, \mathcal{P})$.

Minimization Objective and Risk-Sensitive Control

- A Taylor series expansion of (3) reveals:

$$\mathcal{J}_{exp}(x_0, u, w) = \lim_{T\to\infty} \mathbb{E}\Big|_{x_0\in\mathcal{P}_0} \left[\frac{\alpha}{2}\sum_{t=0}^{T} z^\top(t)z(t)\right] + \frac{\alpha^2}{4} var\left[\sum_{t=0}^{T} z^\top(t)z(t)\right]. \tag{4}$$

- Consider the variance term $\frac{\alpha^2}{4} var\left[\sum_{t=0}^{T} z^\top(t)z(t)\right] \to \epsilon$.
  - α a measure of risk-propensity if $\alpha > 0$;
  - α a measure of risk-aversion if $\alpha < 0$;
  - $\alpha = 0$ implies solving a classic LQP.

RL PO as a Risk-Sensitive Control Problem

- RL (via PG) computes high-variance gradient estimates from Monte-Carlo trajectory roll-outs and bootstrapping.
- If we set $\alpha > 0$ in the LEQG problem (3), we have a controlled setting where we can study the theoretical properties of RL-based PO.
- Framework: an ADP policy iteration (PI) in a continuous PO setting.
- LEQG also interprets as a risk-attenuation algorithm.

Contributions

- A two-loop iterative alternating best-response procedure for computing the optimal mixed-design policy;
- Rigorous convergence analyses follow for the model-based loop updates;
- In the absence of exact system models, we provide an input-to-state-stable hybrid robust stabilization scheme.

Transition Slide

This page is left blank intentionally.

Problem Setup

For $\alpha > 0$, the cost $\mathcal{J}_{exp}(x_0, u) = \mathbb{E}\Big|_{x_0 \in \mathcal{P}_0} \exp\left[\frac{\alpha}{2}\int_0^\infty z^\top(t)z(t)\mathrm{d}t\right]$, becomes

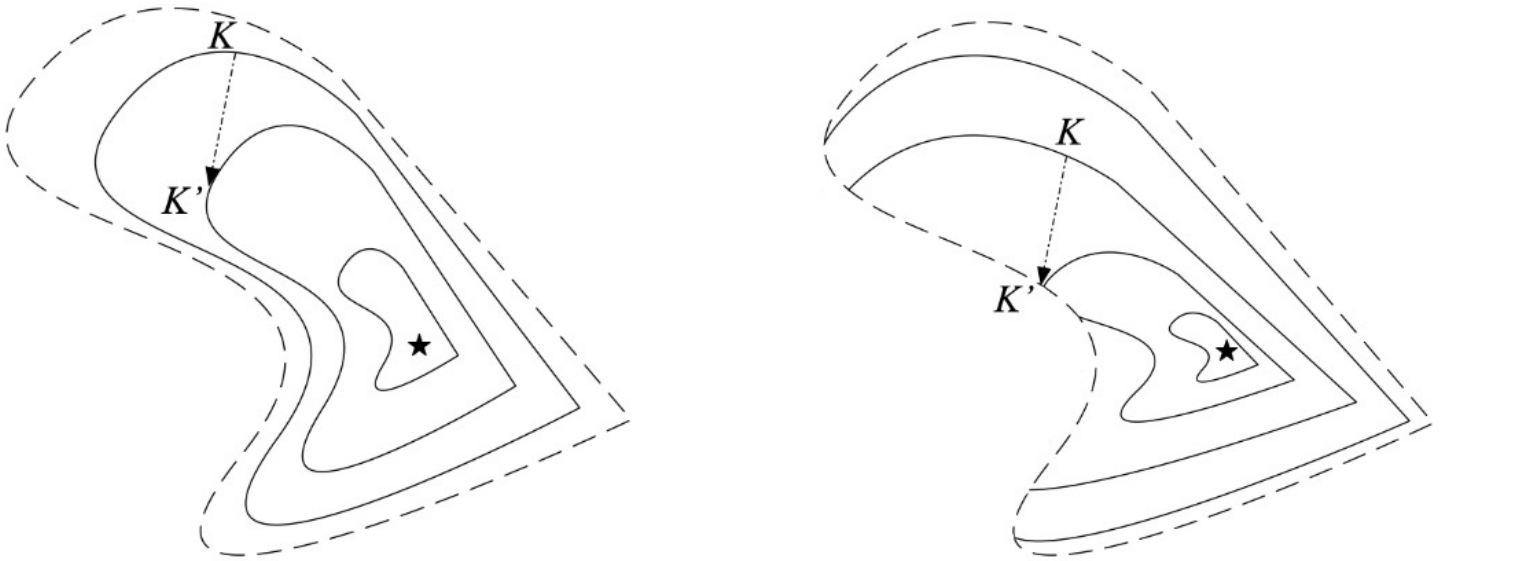
$$\mathbb{E}\Big|_{x_0 \in \mathcal{P}_0} \exp\left\{\frac{\alpha}{2}\int_0^\infty \left[x^\top(t)Qx(t) + u^\top(t)Ru(t)\right]\mathrm{d}t\right\}, \quad (5)$$

with the associated closed loop transfer function,

$$T_{zw}(K) = (C - EK)(sI - A + BK)^{-1}D. \quad (6)$$

Nonconvexity and Coercivity in PG

- Coercivity: iterates remain feasible and strictly separated from the infeasible set as the cost decreases.



(a) Landscape of LQR (b) Landscape of Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control

Figure: Coercivity property of PG on LQR and in mixed-design settings. Credit: (Zhang et al., 2019).

PO and Dynamic Games: Finite-horizon Gain

- Coercivity: feasibility set of optimization iterates

$$\mathcal{K} = \{ K : \ \lambda_i(A - B_1 K) < 0, \ \|T_{zw}(K)\|_\infty < \gamma \}. \quad (7)$$

- Finite-horizon optimization $u^\star(t) = -K^\star_{leqg}\hat{x}(t)$.
- $K^\star_{leqg} = R^{-1}B^\top P_\tau$, and P_τ is the unique, symmetric, positive definite solution to the algebraic Riccati equation (ARE)

$$A^\top P_\tau + P_\tau A - P_\tau(BR^{-1}B^\top - \alpha^{-2}DD^\top)P_\tau = -Q. \quad (8)$$

(Cui and Molu, 2023a, Proposition I), (Duncan, 2013) .

- ∞-horizon case: $P^\star \triangleq P_\infty = \lim_{\tau\to\infty} P_\tau$, and $K^\star_{leqg} \triangleq K_\infty = \lim_{\tau\to\infty} K_\tau$[Theorem on limit of monotonic operators (Kan, 1964)].

Solving the LEQG Problem

- Directly solving the LEQG problem (3) in policy-gradient frameworks incurs biased gradient estimates during iterations;
- Affects risk-sensitivity preservation in infinite-horizon LTI settings (see (Zhang et al., 2021; Zhang et al., 2019));
- Workaround: an equivalent dynamic game formulation to the stochastic LQ PO problem.

Two-Player Zero-Sum Game and LEQG

- An equivalent closed-loop two-player game connection (Cui and Molu, 2023b, Lemma 1):

$$\min_{u\in\mathcal{U}} \max_{\xi\in W} \bar{\mathcal{J}}_\gamma(x_0, u, \xi)$$
$$\text{subject to } \mathrm{d}x(t) = Ax(t)\mathrm{d}t + Bu(t)\mathrm{d}t + D\mathrm{d}w(t),$$
$$z(t) = Cx(t) + Eu(t) \tag{9}$$

$$\bar{\mathcal{J}}_\gamma(x_0, u, \xi) = \mathbb{E}_{x_0\sim\mathcal{P}_0,\,\xi(t)} \int_0^\infty \left[x^\top(t)Qx(t) + u^\top(t)Ru(t)\right] \mathrm{d}t$$
$$-\mathbb{E}_{x_0\sim\mathcal{P}_0,\,\xi(t)} \int_0^\infty \left[\gamma^2\xi^\top(t)\xi(t)\right] \mathrm{d}t$$

, $\xi(\equiv dw) \sim \mathcal{N}(0, \Sigma)$, and $\gamma \equiv \alpha$.

Kleinman's Algorithm

- An iterative algorithm for solving infinite-time Riccati equations (Kleinman, 1968).
- Based on a successive substitution method.
- For a *deterministic LTI system's* cost matrix P_d, the value iterations of P_d^k are monotonically convergent to $P_d^\star$.
- Kleinman's algorithm as policy iteration
  - Choose a stabilizing control gain K_0, and let $p = 0$.
  - (Policy evaluation) Evaluate the performance of K_p from the GARE's solution.
  - (Policy improvement) Improve the policy: $K_p = -R^{-1}B^\top P_d^p$.
  - Advance iteration $p \leftarrow p + 1$.

Model-based Policy Iteration

Algorithm 1: (Model-Based) PO via Policy Iteration

Input: Max. outer iteration $\bar{p}$, $q = 0$, and an $\epsilon > 0$;
Input: Desired risk attenuation level $\gamma > 0$;
Input: Minimizing player's control matrix $R \succ 0$.

Compute $(K_0, L_0) \in \mathcal{K}$; ▷ From [24, Alg. 1];
Set $P^{0,0}_{K,L} = Q^0_K$; ▷ See equation (9);
for $p = 0, \ldots, \bar{p}$ **do**
  Compute Q^p_K and A^p_K ▷ See equation (9);
  Obtain P^p_K by evaluating K_p on (10);
  while $\|P^p_K - P^{p,q}_{K,L}\|_F \leq \epsilon$ **do**
    Compute $L_{q+1}(K_p) := \gamma^{-2} D^\top P^{p,q}_{K,L}$;
    Solve (11) until $\|P^p_K - P^{p,q}_{K,L}\|_F \leq \epsilon$;
    $\bar{q} \leftarrow q + 1$
  end
  Compute $K_{p+1} = R^{-1} B^\top P^{p,\bar{q}}_{K,L}$ ▷ See (11b) ;
end

Outer Loop Convergence: Exponential Stability of P_K^p

Theorem 2

For any $h > 0$ and $K_0 \in \mathcal{K}_h$, there exists $\alpha(h) \in \mathbb{R}$ such that $Tr(P_K^{p+1} - P^\star) \leq \alpha(h) Tr(P_K^p - P^\star)$. That is, $P^\star$ is an exponentially stable equilibrium.

Convergence of the Inner Loop Iteration

Theorem 3

For a $K \in \check{\mathcal{K}}$, and for any $(p, q) \in \mathbb{N}_+$, there exists $\beta(K) \in \mathbb{R}$ such that

$$Tr(P_K^p - P_{K,L}^{p,q+1}) \leq \beta(K)\, Tr(P_K^p - P_{K,L}^{p,q}). \tag{24}$$

Remark 2

As seen from Lemma 5, $P_K^p - P_{K,L}^{p,q} \succeq 0$. By the norm on a matrix trace (Cui and Molu, 2023a, Lemma 13) and the result of Theorem 3, we have $\|P_K - P_{K,L}^{p,q}\|_F \leq Tr(P_K - P_{K,L}^{p,q}) \leq \beta(K)\, Tr(P_K)$, i.e. $P_{K,L}^{p,q}$ exponentially converges to P_K in the Frobenius norm.

Algorithm as a Policy Iteration Scheme

- Choosing a stabilizing K_p we first evaluate u's performance by solving (14).
  - This is the policy evaluation step in PI.
- The policy is then improved in a following iteration by solving for the cost matrix in (15b);
  - This is the policy improvement step.
- Essentially, a policy iteration algorithm whereupon
  - Performance of an initial control gain K_p is first evaluated against a cost function.
  - A newer evaluation of the cost matrix $P_{K,L}^{p,q}$ is then used to improve the controller gain K_{p+1} in the outer loop.

Sampling-based PO: Statement of the Problem

Problem 4 (Sampling-based Policy Optimization)

If A, B, C, D, E, P are all replaced by approximate matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}, \hat{P}$, under what conditions will the sequences $\{\hat{P}^{p,q}_{K,L}\}^{(p,q)=\infty}_{(p,q)=1}$, $\{\hat{K}_p\}^{\infty}_{p=0}$, $\{\hat{L}_q\}^{\infty}_{q=0}$ converge to a small neighborhood of the optimal values $\{P^{\star}_{K,L}\}^{(p,q)=\infty}_{(p,q)=0}$, $\{K^{\star}_p\}^{\infty}_{p=0}$, and $\{L^{\star}_q\}^{\infty}_{q=0}$?

Hybrid System Reparameterization

- Lump estimate errors as an input into the gain terms to be computed in the PO algorithm.
- With inexact outer loop update, K_{p+1} becomes biased so that the inexact outer-loop GARE value iteration involves the recursions

$$\hat{A}_K^{p\top} \hat{P}_K^p + \hat{P}_K^p \hat{A}_K^p + \hat{Q}_K^p + \gamma^{-2} \hat{P}_K^p DD^\top \hat{P}_K^p = 0, \quad (25a)$$

$$\hat{K}_{p+1} = R^{-1} B^\top \hat{P}_K^p + \tilde{K}_{p+1} \triangleq \bar{K}_{p+1} + \tilde{K}_{p+1}, \quad (25b)$$

  - NB: $\hat{A}_K^p = A - B\hat{K}_p$ and $\hat{Q}_K^p = Q + \hat{K}_p^\top R \hat{K}_p$.

Robustness Analyses

- Define $\tilde{P} = P_K - \hat{P}_K$ and $\tilde{K} = K - \hat{K}$.
- Keep $|\tilde{K}| < \epsilon$, start with a $K \in \mathcal{K}$: iterates stay in $\mathcal{K}$.

Lemma 7 (Lemma 10, C&M, '23)

For any $K \in \mathcal{K}$, there exists an $e(K) > 0$ such that for a perturbation $\tilde{K}$, $K + \tilde{K} \in \mathcal{K}$, as long as $\|\tilde{K}\| < e(K)$.

Theorem 6

The inexact outer loop is small-disturbance ISS. That is, for any $h > 0$ and $\hat{K}_0 \in \mathcal{K}_h$, if $\|\tilde{K}\| < f(h)$, there exist a $\mathcal{KL}$-function $\beta_1(\cdot,\cdot)$ and a $\mathcal{K}_\infty$-function $\gamma_1(\cdot)$ such that

$$\|P^p_{\hat{K}} - P^\star\| \leq \beta_1(\|P^0_{\hat{K}} - P^*\|, p) + \gamma_1(\|\tilde{K}\|). \tag{37}$$

Inner Loop Robustness

Theorem 7

Assume $\|\tilde{L}_q(K_p)\| < e$ *for all* $q \in \mathbb{N}_+$. *There exists* $\hat{\beta}(K) \in [0,1)$, *and* $\lambda(\cdot) \in \check{\mathcal{K}}_\infty$, *such that*

$$\|\hat{P}^{p,q}_{K,L} - P^{p,q}_{K,L}\|_F \leq \hat{\beta}^{q-1}(K)\, Tr(P^{p,q}_{K,L}) + \lambda(\|\tilde{L}\|_\infty). \tag{42}$$

- From Theorem 7, as $q \to \infty$, $\hat{P}^{p,q}_{K,L}$ approaches the solution P_K and enters the ball centered at $P^{p,q}_{K,L}$ with radius proportional to $\|\tilde{L}\|_\infty$.
- The proposed inner-loop iterative algorithm well approximates $P^{p,q}_{K,L}$.

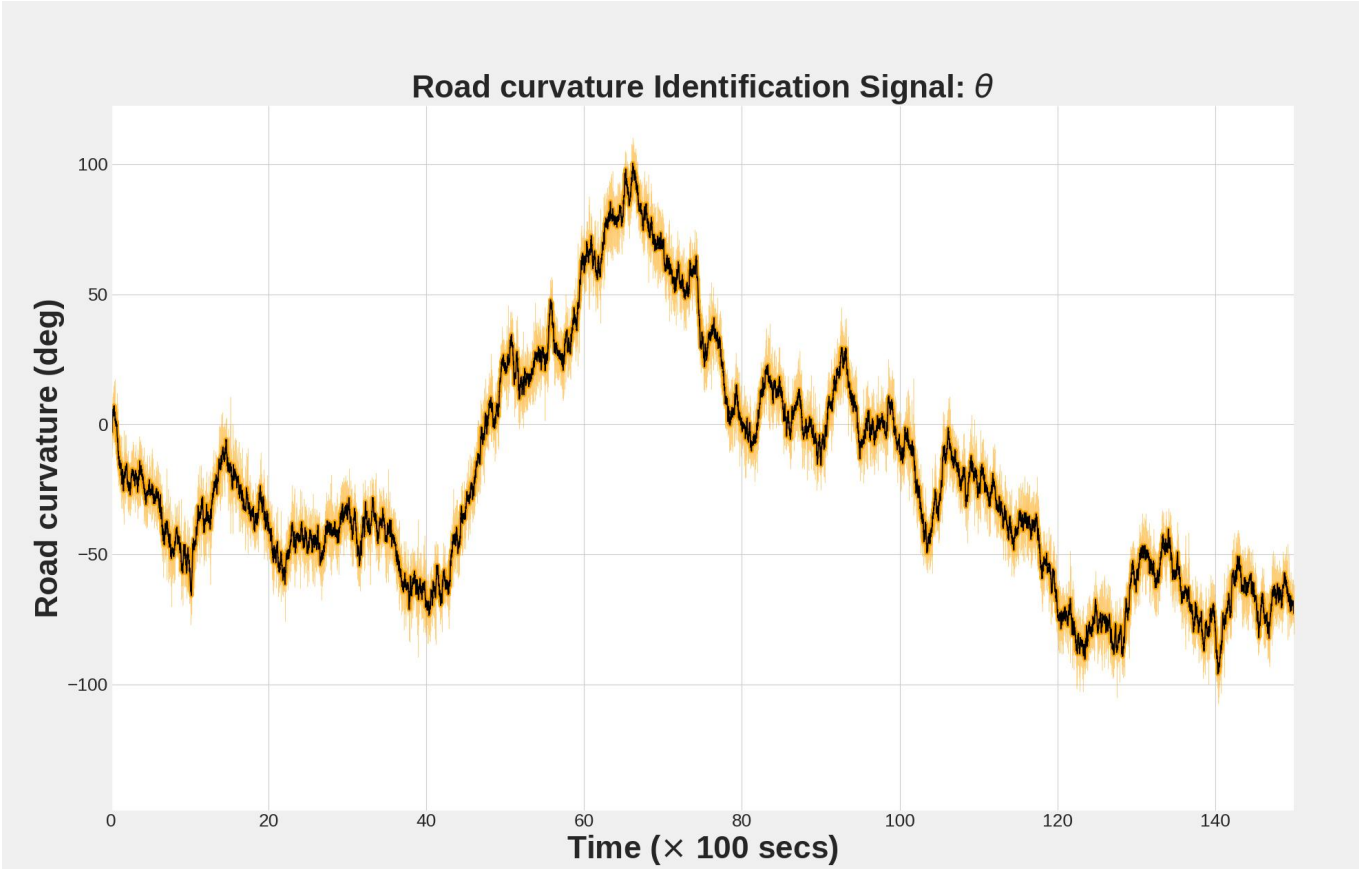
Numerical Results – Car Cruise Control System

- (Åström and Murray, 2021, §3.1):

$$m\frac{dv}{dt} = \alpha_n u \tau(\alpha_n v) - mgC_r sgn(u) - \frac{1}{2}\rho C_d A|v|v - mg\sin\theta \tag{43}$$

- $u(x(t)) = [u_1(t), u_2(t)]$ must maintain a constant velocity v (the state), whilst automatically adjusting the car's throttle, $u_1(t), t \in [0, T]$
  - despite disturbances characterized by road slope changes ($u_3 = \theta$),
  - rolling friction (F_r), and
  - aerodynamic drag forces (F_d).

Road (Disturbance) Profile



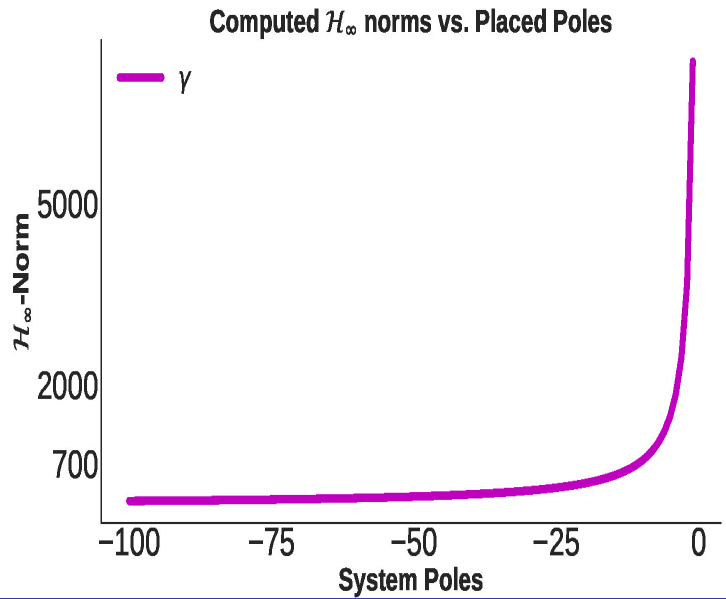
Search for initial stabilizing gain and $\mathcal{H}_\infty$-norm bound.

Proposition 1

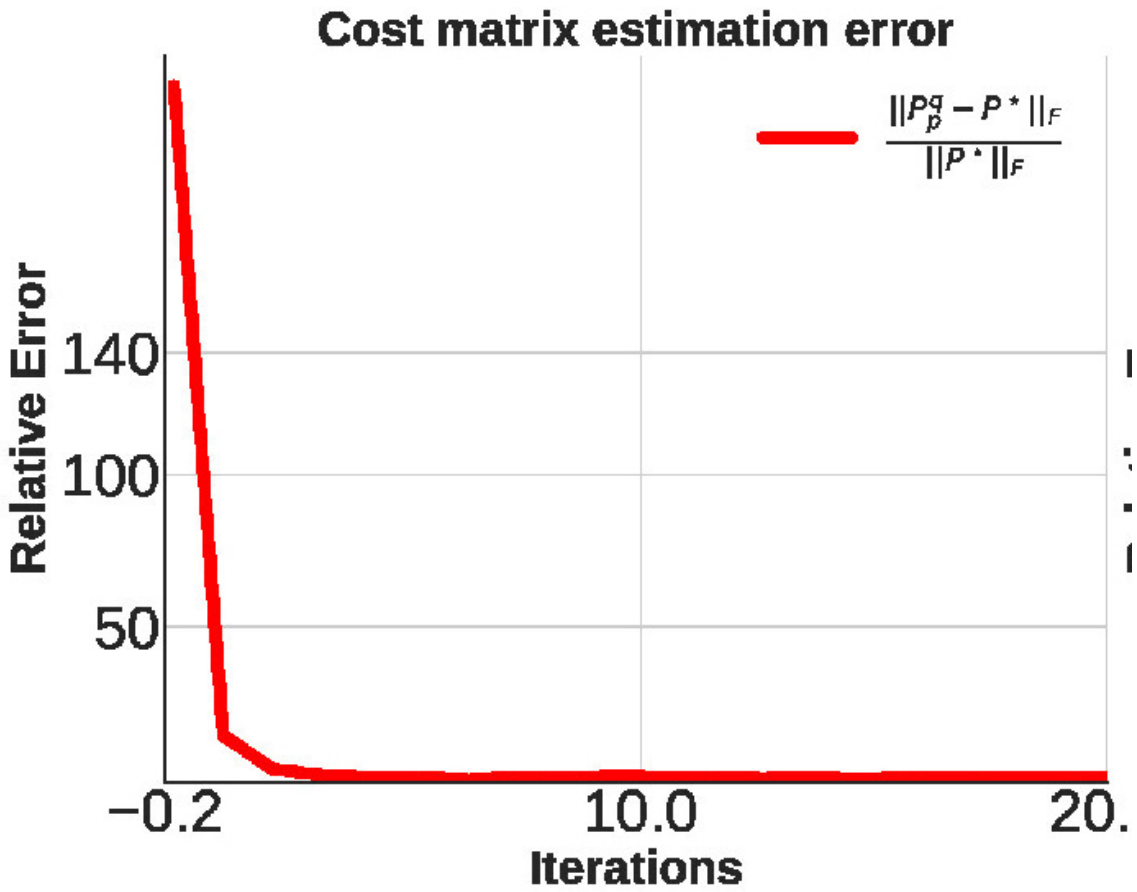
(?) For all $\omega_p \in \mathbb{R}$, we have that $j\omega_p$ is an eigenvalue of the Hamiltonian $H(\gamma_1)$ if and only if γ_1 is a singular value of $T_{zw}(j\omega_p)$.

Algorithm 1 Search for the closed-loop $\mathcal{H}_\infty$-norm

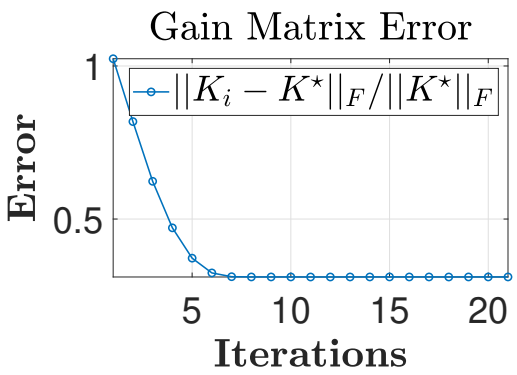
1: Given a user-defined step size $\eta > 0$
2: Set the initial upper bound on γ as $\gamma_{ub} = \infty$.
3: Initialize a buffer for possible $\mathcal{H}_\infty$ norms for each K_1 to be found, $\Gamma_{buf} = \{\}$.
4: Initialize ordered `poles` $\mathcal{P} = \{p_i \in Re(s) < 0 \,|\, i = 1, 2, \}$ ▷ $p_1 < p_2 < \cdots$
5: **for** $p_i \in \mathcal{P}$ **do**
6: Place p_i on (2); ▷ (Tits and Yang, 1996)
7: Compute stabilizing $K_1^{p_i}$
8: Find lower bound γ_{lb} for $H(\gamma, K_1^{p_i})$; ▷ using (22)
9: $\Gamma_{buf}(i) =$ `get_hinf_norm`$(T_{zw}, \gamma_{lb}, K_1^{p_i})$.
10: **end for**
11: **function** `get_hinf_norm`$(T_{zw}, \gamma_{lb}, K_1^{p_i})$
12: **while** $\gamma_{ub} = \infty$ **do**
13: $\gamma := (1 + 2\eta)\,\gamma_{lb}$;
14: Get $\lambda_i(H(\gamma, K_1^{p_i}))$ ▷ c.f. (14)
15: **if** `Re`$(\Lambda) \neq \emptyset$ for $\Lambda = \{\lambda_1, \cdots \lambda_n\}$ **then**
16: Set $\gamma_{ub} = \gamma$; exit
17: **else**
18: Set buffer $\Gamma_{lb} = \{\}$
19: **for** $\lambda_k \in \{$`Imag`$(\Lambda)_{:p-1}\}$ **do** ▷ $k = 1$ to K
20: Set $m_k = \frac{1}{2}(\omega_k + \omega_{k+1})$
21: Set $\Gamma_{lb}(k) = \max\{\sigma\,[T_{zw}(jm_k)]\}$;
22: **end for**
23: $\gamma_{lb} = \max(\Gamma_{lb})$

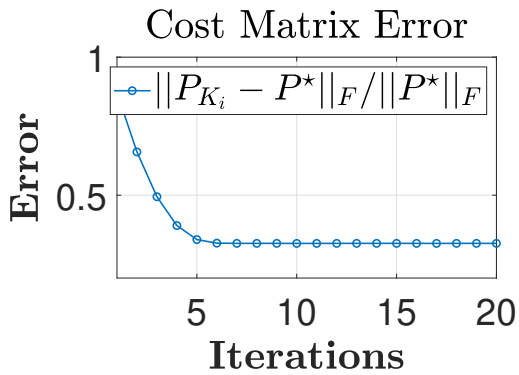


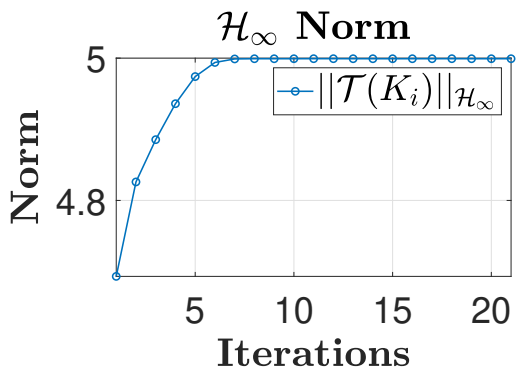
Cost Matrix and Gains Convergence



Pendulums Experiment – Comparison to NPG







Model-free design: $\|\tilde{K}\|_\infty = 0.15$.

Double Pendulum and Acrobot Experiment – Comparison to NPG

Table: Computational Time: Model-based PO vs. Model-free PO vs. NPG.

Policy Optimization Computational time (secs)					
Double Inverted Pendulum			Triple Inverted Pendulum		
Model-based	Model-free	NPG	Model-based	Model-free	NPG
0.0901	0.3061	2.1649	0.1455	0.7829	2.3209

Innovation in the Age of Foundation Models

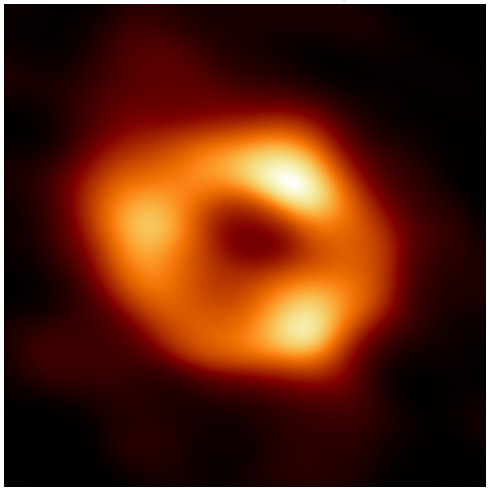
Why am I Here?

If an idea begets a discovery, and if a discovery begets an invention, I am interested in riding the complete **innovation** circuit for intelligence:

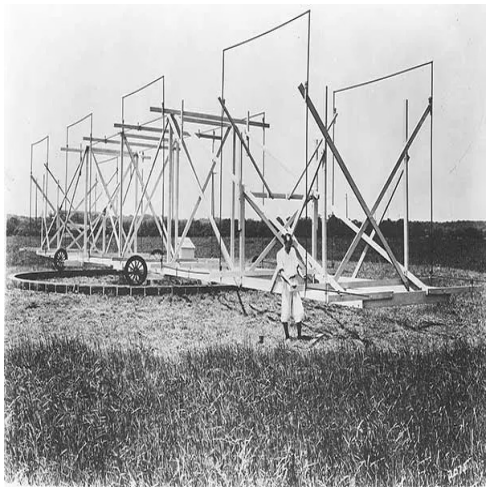
- The thorough and wholesale transformation of fundamental scientific ideas in RL and automation into technological products (or processes) capable of widespread practical use.

Discovery for Physical Autonomy

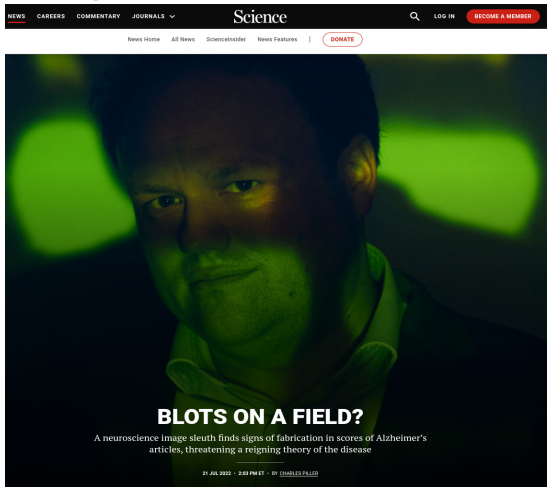
Discovery: The fundamental unit of human progress.



Sagittarius A*, EHT



Karl Jansky, Bell Labs

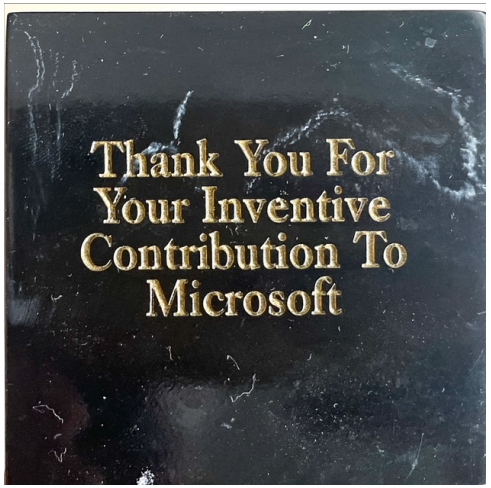


$A\beta^{\star}56$ "undiscovery"

- To wend straight and narrow path between discovery and invention.

Discovery & Invention for Physical Autonomy

Discovery: The fundamental unit of human progress.



Foundation Models, Large Behavior Models

- Large-scale transfer learning, behavior cloning, unsupervised pre-training etc. a new scientific invention.

Credit: π_0: A VLA Flow Model for General Robot Control.

Innovation in the Age of Foundation Models

Why am I Here?

If an idea begets a discovery, and if a discovery begets an invention, I am interested in riding the complete **innovation** circuit for intelligence:

- The thorough and wholesale transformation of fundamental scientific ideas in RL and automation into technological products (or processes) capable of widespread practical use.

Diffusion of Embodied AI

Jack Morton's Corollaries to Innovation

- Three essentials to innovation: "reliability", "reproducibility", and "designability".
- Innovation is a matter of economic imperatives:
  - If you hadn't sold anything you hadn't innovated;
  - Without an affordable price you could never sell anything.

References I

[1] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

[2] Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. *(Accepted for publication at) Conference on Learning Theory*, 2022.

[3] Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=RQLLzMCefQu`.

[4] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[5] Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. *arXiv preprint arXiv:2206.15477*, 2022.

[6] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.