Representation-based Reinforcement Learning

Bo Dai
Google DeepMind & Georgia Tech

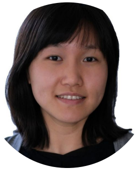
Team

Tongzheng Ren
(UT Austin Alumni)

Chenjun Xiao
(UAlberta)

Tianjun Zhang
(UCB)

Haitong Ma
(Harvard)



Lisa Lee
(Google DeepMind)

Sherry Yang
(Google DeepMind & NYU)

Sujay Sanghavi
(UT Austin)

Na Li
(Harvard)

Csaba Szepesvari
(Google DeepMind & UAlberta)

Dale Schuurmans
(Google DeepMind & UAlberta)

Outline

- Dilemma in RL
  - Difficulties in Model-free and Model-based RL
- An Inspiration from Representation for Control
  - Provable and Practical Stochastic Nonlinear Control
- Coherent Solution: RL with Linear Representation
  - Linear Representation for MDP
  - Linear Representation for POMDP

Markov Decision Processes (MDPs)

Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, T, \mu, H/\gamma \rangle$

- State space: $\mathcal{S}$
- Action space: $\mathcal{A}$
- Reward function: $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition: $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
- Initial state distribution: μ

$$\pi(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{A})$$

$$V_h^\pi(s_h) := \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t, a_t)|s_h = s\right]$$

$$V^\pi(s) := \mathbb{E}_{T,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s\right]$$

$$Q_h^\pi(s_h, a_h) = \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t, a_t)|s_h = s, a_h = a\right]$$

$$Q^\pi(s, a) := \mathbb{E}_{T,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a\right]$$

Markov Decision Processes (MDPs)

Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, T, \mu, H/\gamma \rangle$

- State space: $\mathcal{S}$
- Action space: $\mathcal{A}$
- Reward function: $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition: $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
- Initial state distribution: μ

$$\max_{\pi} J(\pi) = \mathbb{E}_{\mu(s)}[V^{\pi}(s)]$$

$$\pi(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{A})$$

$$V_h^{\pi}(s_h) := \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t, a_t) | s_h = s\right]$$

$$V^{\pi}(s) := \mathbb{E}_{T,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s\right]$$

$$Q_h^{\pi}(s_h, a_h) = \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t, a_t) | s_h = s, a_h = a\right]$$

$$Q^{\pi}(s, a) := \mathbb{E}_{T,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\right]$$

Model-free RL: (deep) Q-Learning

Q-Learning: dynamic programming via Bellman recursion

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \left(T\left(s'|a,s\right) \max_{a'} Q\left(s',a'\right) \right)$$

TD update $$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s',a'\right) - Q_{t-1}(s,a) \right)$$

Deep version $$\theta_t = \theta_{t-1} + \alpha \left(R(s,a) + \gamma \max_{a'} Q_{t-1}(s',a') - Q_{t-1}(s,a) \right) \nabla_\theta Q(s,a)$$

Google

Model-free RL: Policy Gradient

Policy Gradient: direct policy optimization

$$J(\pi_\theta) = \sum_{s \in \mathcal{S}} d^{\pi_\theta}(s) V^{\pi_\theta}(s) = \sum_{s \in \mathcal{S}} d^{\pi_\theta}(s) \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^{\pi_\theta}(s,a)$$

Policy gradient:

$$\nabla_\theta J(\theta) = \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s,a) \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)$$

$$= \mathbb{E}_{T,\pi}\Big[Q^{\pi}(s,a) \nabla_\theta \log \pi_\theta(a|s)\Big]$$

PG update:

$$\theta_t = \theta_{t-1} + \alpha \mathbb{E}_{T,\pi}\Big[Q^{\pi}(s,a) \nabla_\theta \log \pi_\theta(a|s)\Big]$$

Natural PG, Soft AC….

Model-free RL

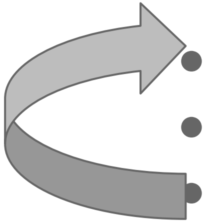
Pros:

- Modeling: easy to incorporate with function approximator, e.g., deep nets, with gradient based learning.

Cons:

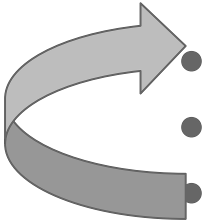
- Exploration: difficulty in capturing the uncertainty with arbitrary nonlinear functions.
- Planning: no guarantee for the global convergence for optimal policy with general nonlinear functions.

Model-based RL



- Collect data through some policy
- Estimate the dynamics model and reward
- Model predictive control based on the estimated models

Model-based RL: LQR



- Collect data through some policy
- Estimate the linear dynamics model and quadratic reward
- Optimize the estimated LQR model

Model-based RL: LQR

Linear Quadratic Regulator

$$\begin{array}{ll} \text{minimize}_{u_t, x_t} & \mathbb{E}\left[\frac{1}{2}\sum_{t=0}^{N}\left\{x_t^T Q x_t + u_t^T R u_t\right\} + \frac{1}{2} x_{N+1}^T S x_{N+1}\right], \\ \text{subject to} & x_{t+1} = A x_t + B u_t + e_t, \text{for } t = 0, 1, \ldots, N, \end{array}$$

With *given* model, we have efficient solution & elegant analysis.

Model-based RL: LQR

Pros:

- Exploration: theoretical-rigorous and computation-efficient uncertainty estimation.
- Planning: elegant planner with global convergence guarantee for solving LQR.

Cons:

- Modeling: linear dynamics model is too restrict.

Model-based RL: Deep MBRL

- Collect data through some policy
- Estimate the dynamics model and reward (deep models)
- Model predictive control based on the estimated parameters

Model-Ensemble Trust-Region Policy Optimization (ME-TRPO)
Stochastic Lower Bound Optimization (SLBO)
Mode-Free Model-Based (MB-MF)
Probabilistic Ensembles with Trajectory Sampling (PETS-RS and PETS-CEM)
[Benchmarking Model-Based Reinforcement Learning]

Deep Model-based RL

Pros:

- Modeling: exploiting the deep models for better approximation.

Cons:

- Exploration: difficulty in capturing the uncertainty with arbitrary nonlinear functions.
- Planning: difficult to control with nonlinear dynamics model.

Dilemma in RL

Trade-off: Modeling, Exploration and Planning

A practical algorithm with rigorous theoretical guarantee to achieve balance?

Representation-based Reinforcement Learning

Representation View for Provable Control

Stochastic Nonlinear Control:

$$\min_{\pi} \qquad \mathbb{E}_{a\sim\pi}\Big[\sum_{h=1}^{H} r(s_h, a_h)\Big]$$

$$s.t. \qquad s_{h+1} = f(s_h, a_h) + \epsilon_h, \quad \text{where} \quad \epsilon_h \sim \mathcal{N}(0, \sigma^2 I)$$

Representation View for Provable Control

Stochastic Nonlinear Control:

$$\min_{\pi} \quad \mathbb{E}_{a\sim\pi}\Big[\sum_{h=1}^{H} r(s_h, a_h)\Big]$$

$$s.t. \quad s_{h+1} = f(s_h, a_h) + \epsilon_h, \quad \text{where} \quad \epsilon_h \sim \mathcal{N}(0, \sigma^2 I)$$

MDP reformulation: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, T, \mu, H \rangle$

$$T(s'|s,a) \propto \exp\Big(-\frac{\|s' - f(s,a)\|_2^2}{2\sigma^2}\Big)$$

Representation View for Provable Control

Stochastic Nonlinear Control:

$$\min_{\pi} \quad \mathbb{E}_{a\sim\pi}\left[\sum_{h=1}^{H} r(s_h, a_h)\right]$$
$$s.t. \quad s_{h+1} = f(s_h, a_h) + \epsilon_h, \quad \text{where} \quad \epsilon_h \sim \mathcal{N}(0, \sigma^2 I)$$

MDP reformulation: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, T, \mu, H \rangle$

$$\begin{aligned} T(s'|s,a) &\propto \exp\Big(-\frac{\|s' - f(s,a)\|_2^2}{2\sigma^2}\Big) \\ &= \langle k(f(s,a), \cdot), k(s', \cdot) \rangle_{\mathcal{H}} \\ &= \langle \phi_\omega(f(s,a)), \phi_\omega(s') \rangle_{\mathcal{N}(0,\sigma^{-2}I)} \end{aligned}$$

$$k(x,y) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$$

$$\phi_\omega(x) = [\cos(\omega^\top x), \sin(\omega^\top x)]$$

Google

Representation View for Provable Control

The transition and reward function are factorizable:

$$T(s'|s,a) = \langle \phi(s,a), \mu(s') \rangle \qquad r(s,a) = \langle \phi(s,a), \theta_r \rangle$$

The value functions defined as

$$V_h^\pi(s_h) := \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t,a_t)|s_h = s\right]$$

$$Q_h^\pi(s_h,a_h) = \mathbb{E}_{T,\pi}\left[\sum_{t=h}^{H-1} r(s_t,a_t)|s_h = s, a_h = a\right]$$

$$Q_h^\pi(s_h,a_h) = r(s_h,a_h) + \mathbb{E}_{s_{h+1}\sim T(\cdot|s_h,a_h)}\left[V_{h+1}^\pi(s_{h+1})\right]$$

Google

Representation View for Provable Control

The transition and reward function are factorizable with $\phi(s, a)$

$$T(s'|s,a) = \langle \phi(s,a), \mu(s') \rangle \qquad r(s,a) = \langle \phi(s,a), \theta_r \rangle$$

Integration representable:

$$\int V_{h+1}^{\pi}(s_{h+1}) T(s_{h+1}|s_h, a_h) \, \mathrm{d}s_{h+1} = \left\langle \phi(s_h, a_h), \int V_{h+1}^{\pi}(s_{h+1}) \mu(s_{h+1}) \, \mathrm{d}s_{h+1} \right\rangle_{\mathcal{H}}.$$

Q-function is linearly representable:

$$\begin{aligned} Q_h^{\pi}(s_h, a_h) &= r(s,a) + \int T(s_{h+1}|s_h, a_h) V_{h+1}^{\pi}(s_{h+1}) ds_{h+1} \\ &= \langle \phi(s,a), \theta_r + \int V_{h+1}^{\pi} \mu(s_{h+1}) ds_{h+1} \rangle_{\mathcal{H}} \end{aligned}$$

Google

Linear MDPs

The transition and reward function are factorizable with $\phi(s, a)$

$$T(s'|s,a) = \langle \phi(s,a), \mu(s') \rangle \qquad r(s,a) = \langle \phi(s,a), \theta_r \rangle$$

Integration representable:

$$\int V^{\pi}_{h+1}(s_{h+1}) T(s_{h+1}|s_h, a_h)\,\mathrm{d}s_{h+1} = \left\langle \phi(s_h, a_h), \int V^{\pi}_{h+1}(s_{h+1}) \mu(s_{h+1})\,\mathrm{d}s_{h+1} \right\rangle_{\mathcal{H}}.$$

Q-function is linearly representable:

$$\begin{aligned} Q^{\pi}_h(s_h, a_h) &= r(s,a) + \int T(s_{h+1}|s_h, a_h) V^{\pi}_{h+1}(s_{h+1}) ds_{h+1} \\ &= \langle \phi(s,a), \theta_r + \int V^{\pi}_{h+1} \mu(s_{h+1}) ds_{h+1} \rangle_{\mathcal{H}} \end{aligned}$$

~~Linear~~ Spectral MDPs

The transition and reward function are factorizable with $\phi(s, a)$

$$T(s'|s,a) = \langle \phi(s,a), \mu(s') \rangle \qquad r(s,a) = \langle \phi(s,a), \theta_r \rangle$$

Integration representable:

Not A Special Model
but a Generic Structure

$$\int V_{h+1}^{\pi}(s_{h+1}) T(s_{h+1}|s_h, a_h)\, \mathrm{d}s_{h+1} = \left\langle \phi(s_h, a_h), \int V_{h+1}^{\pi}(s_{h+1}) \mu(s_{h+1})\, \mathrm{d}s_{h+1} \right\rangle_{\mathcal{H}}.$$

Q-function is linearly representable:

$$\begin{aligned} Q_h^{\pi}(s_h, a_h) &= r(s,a) + \int T(s_{h+1}|s_h, a_h) V_{h+1}^{\pi}(s_{h+1}) ds_{h+1} \\ &= \langle \phi(s,a), \theta_r + \int V_{h+1}^{\pi} \mu(s_{h+1}) ds_{h+1} \rangle_{\mathcal{H}} \end{aligned}$$

Google

Planning for Stochastic Nonlinear Control

Given arbitrary bounded nonlinear transition f, we can construct the representations $\phi(s, a)$ for value function.

Optimization can be solved by dynamic programming in the obtained space.

for steps $h = H-1, H-2, \cdots, 0$ **do**
  Calculate $Q_h(s,a) = r(s,a) + \langle \phi(s,a), \int V_{h+1}(s')\mu(s')\,\mathrm{d}s' \rangle_{\mathcal{H}}$. ▷ Bellman Update.
  Set $V_h(s) = \max_a Q_h(s,a)$, $\pi_h(s) = \arg\max_a Q_h(s,a)$. ▷ Choose the Optimal Policy.
end for

Planning for Stochastic Nonlinear Control

Given arbitrary bounded nonlinear transition f, we can construct the representations $\phi(s, a)$ for value function.

Optimization can be solved by dynamic programming in the obtained space.

for steps $h = H-1, H-2, \cdots, 0$ **do**

$\min_{w_h} \mathbb{E}\left[\left\|w_h^\top \phi(s,a) - r(s,a) - V_{h+1}(s')\right\|^2\right]$ ▷ Bellman Update.

Set $V_h(s) = \max_a Q_h(s,a)$, $\pi_h(s) = \arg\max_a Q_h(s,a)$. ▷ Choose the Optimal Policy.

end for

Google

Thompson Sampling - Exploration vs. Exploitation

Basic idea: pruning the possible model sets with more data observed in a probabilistic way

for episodes $k = 1, 2, \cdots$ **do**
  Sample $f_k \sim \mathbb{P}(f|\mathcal{H}_k)$. ▷ Draw the Representation.
  Find the optimal policy π_k on f_k with Algorithm 2. ▷ Planning with f_k.
  for steps $h = 0, 1, \cdots, H-1$ **do** ▷ Executing π_k.
    Execute $a_h^k \sim \pi_k^h(s_h^k)$.
    Observe s_{h+1}.
  end for
  Set $\mathcal{H}_k = \mathcal{H}_{k-1} \cup \{(s_h^k, a_h^k, s_{h+1}^k)\}_{h=0}^{H-1}$. ▷ Update the History.
end for

Regret Bound

We define the regret of the first K episodes as

$$\mathrm{Regret}(K) := \sum_{k\in[K]} \left[V_0^*(s_0^k) - V_0^{\pi_k}(s_0^k)\right]$$

With some extra assumptions to regularize the transition and reward function, we have

$$\mathbb{E}_{\mathbb{P}(f)}\left[\mathrm{Regret}(K)\right] \le \tilde{\mathcal{O}}(\sqrt{d(\mathcal{F})H^2T})$$

Empirical Performance

	Swimmer	Reacher	MountainCar	Pendulum	I-Pendulum
ME-TRPO*	30.1±9.7	-13.4±5.2	-42.5±26.6	**177.3±1.9**	-126.2±86.6
PETS-RS*	42.1±20.2	-40.1±6.9	-78.5±2.1	167.9±35.8	-12.1±25.1
PETS-CEM*	22.1±25.2	-12.3±5.2	-57.9±3.6	167.4±53.0	-20.5±28.9
DeepSF	25.5±13.5	-16.8±3.6	-17.0±23.4	168.6±5.1	-0.2±0.3
SPEDE	**42.6±4.2**	**-7.2±1.1**	**50.3±1.1**	169.5±0.6	**0.0±0.0**

	Ant-ET	Hopper-ET	S-Humanoid-ET	Humanoid-ET	Walker-ET
ME-TRPO*	42.6±21.1	4.9±4.0	76.1±8.8	72.9±8.9	-9.5±4.6
PETS-RS*	130.0±148.1	205.8±36.5	320.9±182.2	106.9±106.9	-0.8±3.2
PETS-CEM*	81.6±145.8	129.3±36.0	355.1±157.1	110.8±91.0	-2.5±6.8
DeepSF	768.1±44.1	548.9±253.3	533.8±154.9	168.6±5.1	165.6±127.9
SPEDE	**806.2±60.2**	**732.2±263.9**	**986.4±154.7**	**886.9±95.2**	**501.6±204.0**

Empirical Performance

	Swimmer	Reacher	MountainCar	Pendulum	I-Pendulum
PPO*	38.0±1.5	-17.2±0.9	27.1±13.1	163.4±8.0	-40.8±21.0
TRPO*	37.9±2.0	-10.1±0.6	-37.2±16.4	166.7±7.3	-27.6±15.8
TD3*	40.4±8.3	-14.0±0.9	-60.0±1.2	161.4±14.4	-224.5±0.4
SAC*	**41.2±4.6**	-6.4±0.5	**52.6±0.6**	168.2±9.5	-0.2±0.1
SPEDE-REG	40.0±3.8	**-5.8±0.6**	40.0±3.8	**168.5±4.3**	**0.0±0.1**

	Ant-ET	Hopper-ET	S-Humanoid-ET	Humanoid-ET	Walker-ET
PPO*	80.1±17.3	758.0±62.0	454.3±36.7	451.4±39.1	306.1±17.2
TRPO*	116.8±47.3	237.4±33.5	281.3±10.9	289.8±5.2	229.5±27.1
TD3*	259.7±1.0	1057.1±29.5	1070.0±168.3	147.7±0.7	**3299.7±1951.5**
SAC*	**2012.7±571.3**	1815.5±655.1	834.6±313.1	1794.4±458.3	2216.4±678.7
SPEDE-REG	**2073.1±119.7**	**2510.3±550.8**	**2710.3±277.5**	**3747.8±1078.1**	2170.3±810.9

Summary and Gaps

Take home message:

- Linearization makes nonlinear potentially solvable
- Linearization bridges model-free and model-based RL

Gaps between theory vs. practice:

- Infinite-dim linearization approximation (Ren et al, CDC 2023)
- Posterior approximation
- Gaussian noise

Could we do better to avoid these limitations?

Learning Single Feature for Linear MDPs

- Un-normalized conditional density: intractable MLE

$$\max_{\phi,\mu} \widehat{\mathbb{E}}_{s,a,s'}\left[\log\langle\phi(s,a),\mu(s')\rangle\right]$$
$$\texttt{s.t.}\ \langle\phi(s,a),\mu(s')\rangle = 1, \quad \forall(s,a)\in\mathcal{S}\times\mathcal{A}$$

- Feature is changing: exploration in a nonlinear space, is UCB still working?

Learning Single Feature for Linear MDPs

- Un-normalized conditional density: intractable MLE

$$\max_{\phi,\mu} \widehat{\mathbb{E}}_{s,a,s'} \left[\log\langle \phi(s,a), \mu(s') \rangle \right]$$

$$\texttt{s.t.} \; \langle \phi(s,a), \mu(s') \rangle = 1, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

- Feature is changing: exploration in a nonlinear space, is UCB still working?

Google

Uehara, M., Zhang, X., & Sun, W. (2021). Representation Learning for Online and Offline RL in Low-rank MDPs. *arXiv preprint arXiv:2110.04652.*

Alternative?

- Un-normalized conditional density

$$T(s'|s,a) = \frac{\langle \phi(s,a), \mu(s') \rangle}{Z(s,a)}, \quad Z(s,a) = \int \langle \phi(s,a), \mu(s') \rangle ds'$$

$$\max_{\phi,\mu} \widehat{\mathbb{E}}_{s,a,s'} \left[\log \langle \phi(s,a), \mu(s') \rangle \right] - \log Z(s,a)$$

Induce difficulty in representing $$Q(s,a) = \left\langle \underline{\frac{\phi(s,a)}{Z(s,a)}}, w \right\rangle$$

Google

Making Linear Representations Learning Tractable

- We consider a contrastive loss (NCE/CPC) as a tractable alternative to the MLE

$$(s, a, s') \sim \mathcal{D}, \quad s_l \sim p(s')$$

$$\max_{\phi,\mu} \hat{\mathbb{E}}\left[\langle \phi(s,a), \mu(s') \rangle - \log \sum_l \langle \phi(s,a), \mu(s'_l) \rangle\right]$$

Making Linear Representations Learning Tractable

- We consider a contrastive loss (NCE/CPC) as a tractable alternative to the MLE

$$(s, a, s') \sim \mathcal{D}, \quad s_l \sim p(s')$$

$$\max_{\phi,\mu} \hat{\mathbb{E}}\left[\langle \phi(s,a), \mu(s')\rangle - \log \sum_l \langle \phi(s,a), \mu(s_l')\rangle\right]$$

We can show the objective leads to solution

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s')\rangle$$

Google

Making Linear Representations Learning Tractable

- We consider the SVD as a tractable alternative to the MLE

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s') \rangle \quad \Rightarrow \frac{T(s',s,a)}{\sqrt{p(s,a)}\sqrt{p(s')}} = \sqrt{p(s,a)}\sqrt{p(s')}\phi(s,a)^\top \mu(s')$$

SVD decomposition

$$\int \left\| \frac{T(s',s,a)}{\sqrt{p(s,a)}\sqrt{p(s')}} - \sqrt{p(s,a)}\sqrt{p(s')}\phi(s,a)^\top \mu(s') \right\|^2 d(s,a)ds'$$

Google

Making Linear Representations Learning Tractable

- We consider the SVD as a tractable alternative to the MLE

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s') \rangle \quad \Rightarrow \frac{T(s',s,a)}{\sqrt{p(s,a)}\sqrt{p(s')}} = \sqrt{p(s,a)}\sqrt{p(s')}\phi(s,a)^\top \mu(s')$$

SVD decomposition

$$\int \left\| \frac{T(s',s,a)}{\sqrt{p(s,a)}\sqrt{p(s')}} - \sqrt{p(s,a)}\sqrt{p(s')}\phi(s,a)^\top \mu(s') \right\|^2 d(s,a)ds'$$

$$\propto -2\mathbb{E}_{T(s',s,a)}[\phi(s,a)^\top \mu(s')] + \mathbb{E}_{p(s,a)p(s')}[(\phi(s,a)^\top \mu(s'))^2]$$

Google

Making Linear Representations Learning Tractable

- We connect the Latent Variable Model with Linear MDP

$$T(s'|s,a) = \int p(s'|z)p(z|s,a)dz = \langle p(z|s,a), p(s'|z)\rangle_{L_2}$$

$$\phi(s,a) = p(z|s,a), \quad \mu(s') = p(s'|z)$$

Making Linear Representations Learning Tractable

- We connect the Latent Variable Model with Linear MDP

$$T(s'|s,a) = \int p(s'|z)p(z|s,a)dz = \langle p(z|s,a), p(s'|z)\rangle_{L_2}$$

$$\phi(s,a) = p(z|s,a), \quad \mu(s') = p(s'|z)$$

$$Q(s,a) = \int w(z)p(z|s,a)dz$$

Making Linear Representations Learning Tractable

- We connect the Latent Variable Model with Linear MDP

$$T(s'|s,a) = \int p(s'|z)p(z|s,a)dz = \langle p(z|s,a), p(s'|z)\rangle_{L_2}$$

Evidence Lower Bound (ELBO) of LVM

$$\begin{aligned}\log T(s'|s,a) &= \log \int p(s'|z)p(z|s,a)dz \\ &= \max_{q\in\Delta(\mathcal{Z})} \mathbb{E}_q[\log p(s'|z)] - KL(q(z|s,a,s')||p(z|s,a))\end{aligned}$$

Making Linear Representations Learning Tractable

- We connect the Diffusion Model with spectral decomposition MDP

$$T(s'|s,a) \propto \exp(\psi(s,a)^\top \upsilon(s')) = \langle \phi_\omega(\psi(s,a)), \nu_\omega(\upsilon(s')) \rangle$$

Score-base Representation Learning

$$\min_{\psi,\upsilon} \; \mathbb{E}_\beta \mathbb{E}_{s,a,s',\tilde{s}'} \left[\| \tilde{s}' + \beta \psi(s,a)^\top \nabla_{\tilde{s}'} \upsilon(\tilde{s}', \beta) - \sqrt{1-\beta} s' \|^2 \right]$$

Google

Algorithm

- Collect data $s \sim d^{\pi_n}, a \sim U(\mathcal{A}), s' \sim T(\cdot|s,a)$
- $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{s, a, s'\}$
- Learn representation via NCE / SVD / ELBO
- Calculate UCB bonus $\widehat{b}_n(s,a) = \alpha_n \sqrt{\widehat{\phi}_n(s,a)^\top \widehat{\Sigma}_n^{-1} \widehat{\phi}_n(s,a)}$ $\quad \widehat{\Sigma}_n = \sum_{s,a \in \mathcal{D}_n} \widehat{\phi}_n(s,a) \widehat{\phi}_n(s,a)^\top + \lambda_n I$
- Policy evaluation with Bellman recursion

$$Q^\pi(s,a) = r(s,a) + \widehat{b}_n(s,a) + \gamma \mathbb{E}_P \left[V^\pi(s') \right]$$

- Policy Optimization with learned Q

Sample complexity $\texttt{poly}\left(d, |\mathcal{A}|, \frac{1}{(1-\gamma)}/H, \epsilon\right)$ such that $V_{P,r}^{\pi^*} - V_{P,r}^{\pi} \leqslant \epsilon$

Google

Empirical Performances

		HalfCheetah	Reacher	Humanoid-ET	Pendulum	I-Pendulum
Model-Based RL	ME-TRPO*	2283.7±900.4	-13.4±5.2	72.9±8.9	**177.3±1.9**	-126.2±86.6
	PETS-RS*	966.9±471.6	-40.1±6.9	109.6±102.6	167.9±35.8	-12.1±25.1
	PETS-CEM*	2795.3±879.9	-12.3±5.2	110.8±90.1	167.4±53.0	-20.5±28.9
	Best MBBL	3639.0±1135.8	**-4.1±0.1**	1377.0±150.4	**177.3±1.9**	**0.0±0.0**
Model-Free RL	PPO*	17.2±84.4	-17.2±0.9	451.4±39.1	163.4±8.0	-40.8±21.0
	TRPO*	-12.0±85.5	-10.1±0.6	289.8±5.2	166.7±7.3	-27.6±15.8
	SAC* (3-layer)	4000.7±202.1	-6.4±0.5	**1794.4±458.3**	168.2±9.5	-0.2±0.1
Representation RL	DeepSF	4180.4±113.8	-16.8±3.6	168.6±5.1	168.6±5.1	-0.2±0.3
	SPEDE	4210.3±92.6	-7.2±1.1	886.9±95.2	169.5±0.6	0.0±0.0
	SPEDER	**5407.9±813.0**	-5.90±0.3	1774.875±129.1	167.4±3.4	**0.0±0.0**
		Ant-ET	Hopper-ET	S-Humanoid-ET	CartPole	Walker-ET
Model-Based RL	ME-TRPO*	42.6±21.1	1272.5±500.9	-154.9±534.3	160.1±69.1	-1609.3±657.5
	PETS-RS*	130.0±148.1	205.8±36.5	320.7±182.2	195.0±28.0	312.5±493.4
	PETS-CEM*	81.6±145.8	129.3±36.0	355.1±157.1	195.5±3.0	260.2±536.9
	Best MBBL	275.4±309.1	1272.5±500.9	**1084.3±77.0**	200.0±0.0	312.5±493.4
Model-Free RL	PPO*	80.1±17.3	758.0±62.0	454.3±36.7	86.5±7.8	306.1±17.2
	TRPO*	116.8±47.3	237.4±33.5	281.3±10.9	47.3±15.7	229.5±27.1
	SAC* (3-layer)	2012.7±571.3	1815.5±655.1	834.6±313.1	199.4±0.4	2216.4±678.7
Representation RL	DeepSF	768.1±44.1	548.9±253.3	533.8±154.9	194.5±5.8	165.6±127.9
	SPEDE	806.2±60.2	732.2±263.9	986.4±154.7	138.2±39.5	501.6±204.0
	SPEDER	**1806.8±1488.0**	**2267.6±554.3**	944.8±354.3	**200.2±1.0**	**2451.5±1115.6**

Empirical Performances

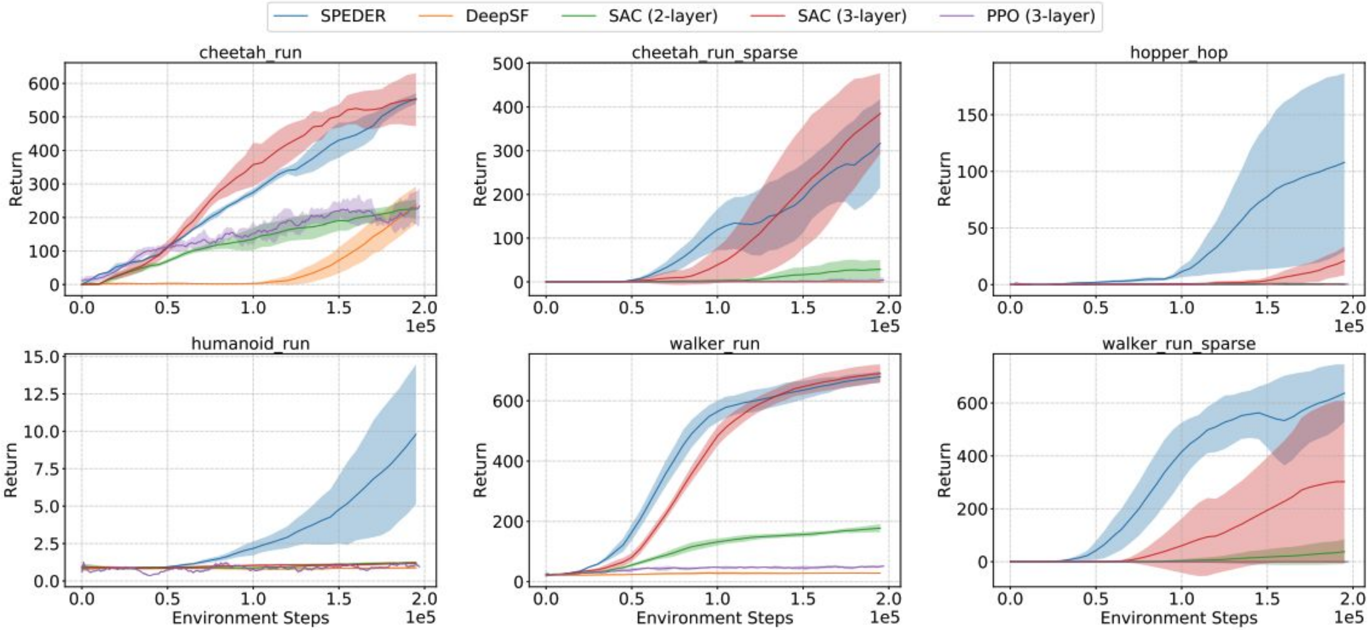


Figure 4: Performance Curves for online DM Control Suite.

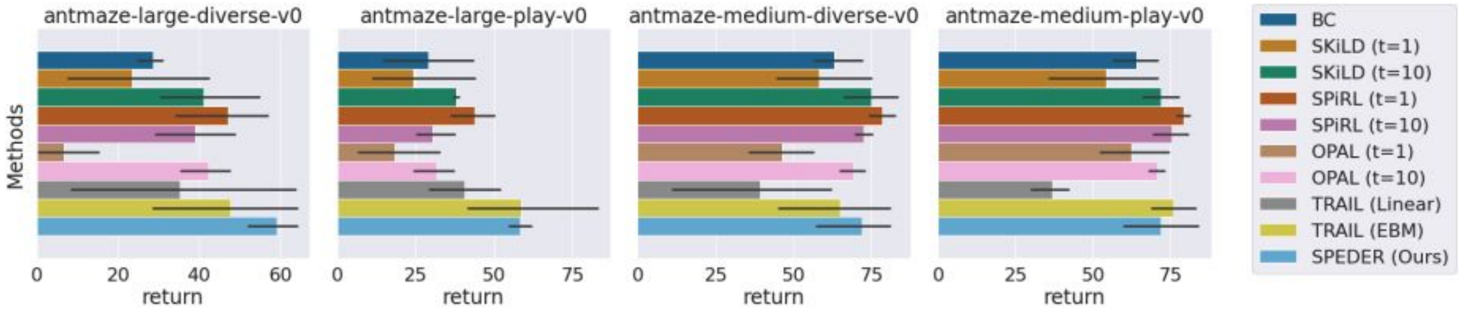
Google

Representations vs. Skills Learning

Correspondence between policies and value functions

$$\pi_Q(a|s) := \frac{\exp(Q(s,a))}{\sum_{a\in\mathcal{A}}\exp(Q(s,a))} = \mathop{\arg\max}_{\pi(\cdot|s)\in\Delta(\mathcal{A})} \mathbb{E}_\pi\left[Q(s,a)\right] + H(\pi),$$

$\phi(s,a)$ forms value functions, therefore, induces skills.



Google

Byproduct of the Reference Distribution

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s') \rangle$$

Stationary Occupancy Distribution in infinite-horizon MDP

$$\begin{aligned} d^\pi(s) &= (1-\gamma)\mu_0(s) + \gamma \int T(s|s',a')d^\pi(s')\pi(a'|s')ds'da' \\ &= (1-\gamma)\mu_0(s) + \gamma \langle p(s)\mu(s'), \int \phi(s',a')d^\pi(s')\pi(a'|s')ds'da' \rangle \end{aligned}$$

Google

Byproduct of the Reference Distribution

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s') \rangle$$

Stationary Occupancy Distribution in infinite-horizon MDP

$$d^\pi(s) = (1-\gamma)\mu_0(s) + \gamma \int T(s|s',a')d^\pi(s')\pi(a'|s')ds'da'$$

$$= (1-\gamma)\mu_0(s) + \gamma\langle p(s)\mu(s'), \int \phi(s',a')d^\pi(s')\pi(a'|s')ds'da' \rangle$$

$$\frac{d^\pi(s)}{p(s)} = (1-\gamma)\frac{\mu_0(s)}{p(s)} + \gamma\langle \mu(s'), \int \phi(s',a')d^\pi(s')\pi(a'|s')ds'da' \rangle$$

Linear Stationary Ratio

Primal-Dual Spectral Representation in DICE

$$T(s'|s,a) = \langle \phi(s,a), p(s')\mu(s') \rangle$$

Stationary Occupancy Distribution in infinite-horizon MDP

$$\begin{aligned} d^\pi(s) &= (1-\gamma)\mu_0(s) + \gamma \int T(s|s',a')d^\pi(s')\pi(a'|s')ds'da' \\ &= (1-\gamma)\mu_0(s) + \gamma \langle p(s)\mu(s'), \int \phi(s',a')d^\pi(s')\pi(a'|s')ds'da' \rangle \end{aligned}$$

$$\frac{d^\pi(s)}{p(s)} = (1-\gamma)\frac{\mu_0(s)}{p(s)} + \gamma \langle \mu(s'), \int \phi(s',a')d^\pi(s')\pi(a'|s')ds'da' \rangle$$

Linear Stationary Ratio

Yang Hu, Tianyi Chen, Na Li, Kai Wang, Bo Dai. Primal-Dual Spectral Representation for Off-policy Evaluation. ArXiv, 2024

Summary and Gaps

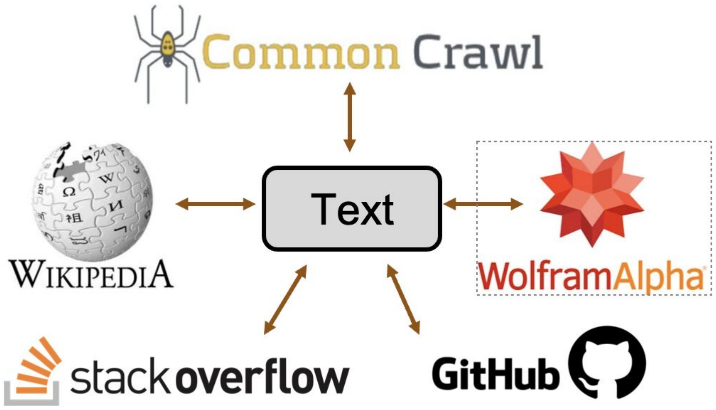
Linearization enables RL with nonlinear models:

- efficient exploration
- efficient planning

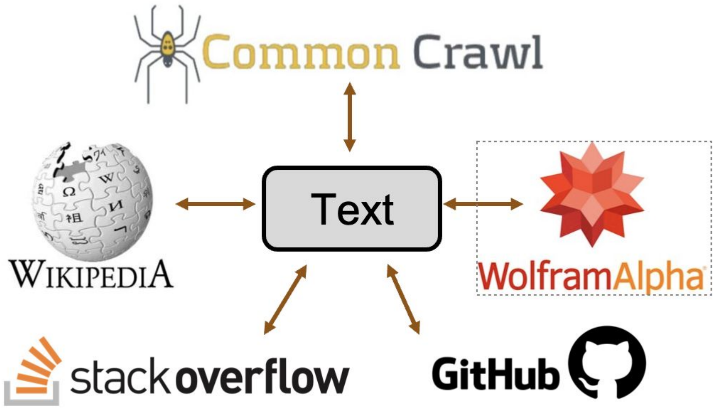
Still not applicable for practical setting:

- RL from observations, e.g., images/videos/texts

Rich Observations in Real World



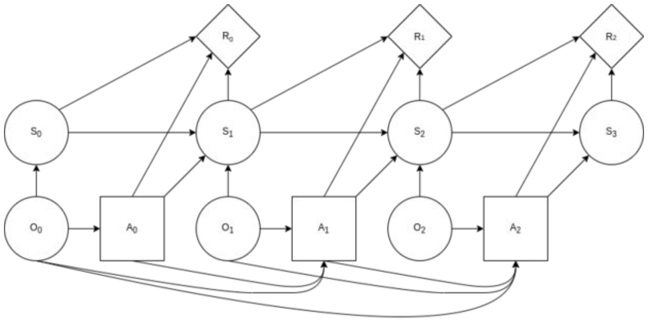
Rich Observations in Real World



But no complete state information

Linear Representation for POMDPs

Partially Observable MDP $\mathcal{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, r, H, \rho_0, T, O \rangle$



pomdp

More drafts

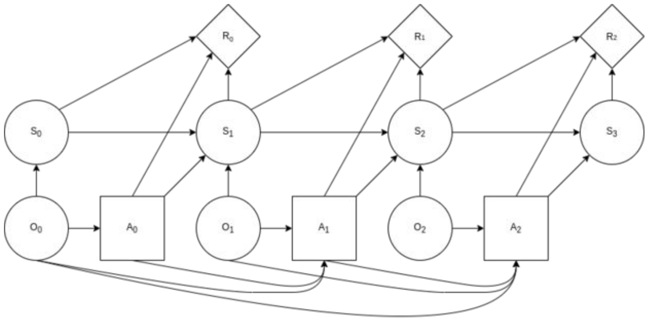
A partially observable Markov decision process (POMDP) is a mathematical framework for modeling decision-making problems where the agent has limited information about the state of the environment. It is a generalization of a Markov decision process (MDP), where the agent has complete knowledge of the state.

In a POMDP, the agent can only observe the environment indirectly through noisy observations. The agent must then make decisions based on these observations, trying to maximize its expected reward over time.

POMDPs have a wide range of applications, including robotics, planning, control, finance, and healthcare. For example, a POMDP could be used to model a robot that is trying to navigate an unknown environment, or a doctor who is trying to diagnose a patient based on their symptoms.

POMDPs are difficult, but NOT all of them

Partially Observable MDP $\mathcal{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, r, H, \rho_0, T, O \rangle$



Computation: PSPACE-complete (Papadimitriou & Tsitsiklis, 1987)
Statistic: Exponentially w.r.t. the horizon (Jin et al., 2020a)

Structured POMDPs with efficient sample complexity (Jin et al., 2020a; Golowich et al., 2022; Liu et al., 2022; 2023, Efroni et al., 2022; Guo et al., 2023)

The Difficulty of POMDPs

Equivalent Beliefs MDPs

$$b(s_{h+1}|\tau_{h+1}) \propto \int_{\mathcal{S}} b(s_h|\tau_h)\mathbb{P}(s_{h+1}|s_h, a_h)\mathbb{O}(o_{h+1}|s_{h+1})ds_h$$

$$Q_h^\pi(b_h, a_h) = r(o_h, a_h) + \mathbb{E}_{b_h(s)}\left[\int \mathbb{P}(s_{h+1}|s_h, a_h)\mathbb{E}_{O(o_{h+1}|s_{h+1})}\left[V_{h+1}^\pi(b(\tau_h, a_h, o_{h+1}))\right]\right]$$

The Difficulty of POMDPs

Equivalent Beliefs MDPs

$$b(s_{h+1}|\tau_{h+1}) \propto \int_{\mathcal{S}} b(s_h|\tau_h) T(s_{h+1}|s_h, a_h) \mathbb{O}(o_{h+1}|s_{h+1}) ds_h$$

$$Q_h^{\pi}(b_h, a_h) = r(o_h, a_h) + \mathbb{E}_{b_h(s)} \left[\int T(s_{h+1}|s_h, a_h) \mathbb{E}_{\mathbb{O}(o_{h+1}|s_{h+1})} \left[V_{h+1}^{\pi}(b(\tau_h, a_h, o_{h+1})) \right] \right]$$

L-decodable POMDPs

Definition 1 (L-decodability [Efroni et al., 2022]) $\forall h \in [H]$, *define*

$$x_h \in \mathcal{X} := (\mathcal{O} \times \mathcal{A})^{L-1} \times \mathcal{O},$$
$$x_h = (o_{h-L+1}, a_{h-L+1}, \cdots, o_h).$$

A POMDP is L-decodable if there exists a decoder $p^* : \mathcal{X} \to \Delta(\mathcal{S})$ such that $p^*(x_h) = b(\tau_h)$.

Linear Representation for POMDPs

$$Q_h^\pi(x_h, a_h) = r(o_h, a_h) + \mathbb{E}_{\mathbb{P}^\pi(o_{h+1}|x_h, a_h)}\left[V_{h+1}^\pi(x_{h+1})\right].$$

$$x_{h+1} = (o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h, o_{h+1})$$

$$(x_h, a_h) = (o_{h-L+1}, a_{h-L+1}, o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h).$$

Linear Representation for POMDPs

$$Q_h^\pi(x_h, a_h) = r(o_h, a_h) + \mathbb{E}_{\mathbb{P}^\pi(o_{h+1}|x_h, a_h)}\left[V_{h+1}^\pi(x_{h+1})\right].$$

$$\int \mathbb{P}^\pi(o_{h+1}|x_h, a_h) V_{h+1}^\pi(o_{h-L+2}, a_{h-L+2}, \ldots, o_h, a_h, o_{h+1}) do_{h+1}$$

$$x_{h+1} = (o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h, o_{h+1})$$

$$(x_h, a_h) = (o_{h-L+1}, a_{h-L+1}, o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h).$$

Linear Representation for POMDPs

Under L-Step Decodable Assumption

$$Q_h^\pi(x_h, a_h) = \mathbb{E}_{\tau_{h+1:h+L}|x_h, a_h}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + V_{h+L}^\pi(x_{h+L})\right]$$

Under Moment Matching Policy

$$\mathbb{P}^\pi(x_{h+L}|x_h, a_h) = \int p(z_{h+1}|x_h, a_h)\mathbb{P}^{\nu_\pi}(x_{h+L}|z_{h+1})\, dz_{h+1} = \langle p(\cdot|x_h, a_h), \mathbb{P}^{\nu_\pi}(x_{h+L}|\cdot)\rangle_{L_2(\mu)}$$

Linear Representation for POMDPs

Under L-Step Decodable Assumption

$$Q_h^\pi(x_h, a_h) = \mathbb{E}_{\tau_{h+1:h+L}|x_h, a_h}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + V_{h+L}^\pi(x_{h+L})\right]$$

$$\mathbb{E}^\pi_{o_{h+k}|x_h, a_h}\left[r\left(o_{h+k}, a_{h+k}\right)\right] = \left\langle p(\cdot|x_h, a_h), \underbrace{\int \mathbb{P}^{\nu_\pi}\left(o_{h+k}, a_{h+k}|\cdot\right) r\left(o_{h+k}, a_{h+k}\right) do_{h+k} da_{h+k}}_{w_k^\pi(\cdot)}\right\rangle$$

Linear Representation for POMDPs

Under L-Step Decodable Assumption

$$Q_h^\pi(x_h, a_h) = \mathbb{E}_{\tau_{h+1:h+L}|x_h, a_h}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + V_{h+L}^\pi(x_{h+L})\right]$$

$$\mathbb{E}^\pi_{o_{h+k}|x_h, a_h}\left[r\left(o_{h+k}, a_{h+k}\right)\right] = \left\langle p(\cdot|x_h, a_h), \underbrace{\int \mathbb{P}^{\nu_\pi}\left(o_{h+k}, a_{h+k}|\cdot\right) r\left(o_{h+k}, a_{h+k}\right) do_{h+k} da_{h+k}}_{w_k^\pi(\cdot)}\right\rangle$$

$$\mathbb{E}_\pi\left[V_{h+L}^\pi\left(x_{h+L}\right)\right] = \int \mathbb{P}^\pi\left(x_{h+L}|x_h, a_h\right) V^\pi\left(x_{h+L}\right) dx_{h+L} = \left\langle p(\cdot|x_h, a_h), \underbrace{\int \mathbb{P}^{\nu_\pi}\left(x_{h+L}|\cdot\right) V^\pi\left(x_{h+L}\right) dx_{h+L}}_{w_{h+L}^\pi(\cdot)}\right\rangle$$

Linear Representation for POMDPs

Under L-Step Decodable Assumption

$$Q_h^\pi(x_h, a_h) = \mathbb{E}_{\tau_{h+1:h+L}|x_h,a_h}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + V_{h+L}^\pi(x_{h+L})\right]$$

$$Q_h^\pi(x_h, a_h) = \langle p(\cdot|x_h, a_h), w^\pi(\cdot)\rangle_{L_2(\mu)}$$

Google

Linear Representation for POMDPs

$$
\begin{aligned}
&\log p\left(o_{h+1:h+l} | x_h, a_h\right) = \log \int_{\mathcal{Z}} p(z_h | x_h, a_h) \mathbb{P}^\pi\left(o_{h+1:h+l} | z_h\right) \\
&= \log \int_{\mathcal{Z}} \frac{p(z_h | x_h, a_h) \mathbb{P}^\pi\left(o_{h+1:h+l} | z_h\right)}{q(z | x_h, a_h, o_{h+1:h+l})} q(z | x_h, a_h, o_{h+1:h+l}) \qquad (17) \\
&= \max_{q \in \Delta(\mathcal{Z})} \mathbb{E}_{q(\cdot | x_h, a_h, o_{h+1:h+l})}\left[\log \mathbb{P}^\pi\left(o_{h+1:h+l} | z_h\right)\right] - D_{KL}\left(q(z | x_h, a_h, o_{h+1:h+l}) || p(z_h | x_h)\right),
\end{aligned}
$$

Linear Representation for POMDPs

$$
\begin{aligned}
\log p\left(o_{h+1:h+l} \middle| x_h, a_h\right) &= \log \int_{\mathcal{Z}} p(z_h|x_h, a_h)\mathbb{P}^{\pi}\left(o_{h+1:h+l} \middle| z_h\right) \\
&= \log \int_{\mathcal{Z}} \frac{p(z_h|x_h, a_h)\mathbb{P}^{\pi}\left(o_{h+1:h+l}|z_h\right)}{q(z|x_h, a_h, o_{h+1:h+l})} q(z|x_h, a_h, o_{h+1:h+l}) \\
&= \max_{q \in \Delta(\mathcal{Z})} \mathbb{E}_{q(\cdot|x_h, a_h, o_{h+1:h+l})}\left[\log \mathbb{P}^{\pi}\left(o_{h+1:h+l}|z_h\right)\right] - D_{KL}\left(q(z|x_h, a_h, o_{h+1:h+l}) \| p(z_h|x_h)\right),
\end{aligned} \tag{17}
$$

A Special World Model

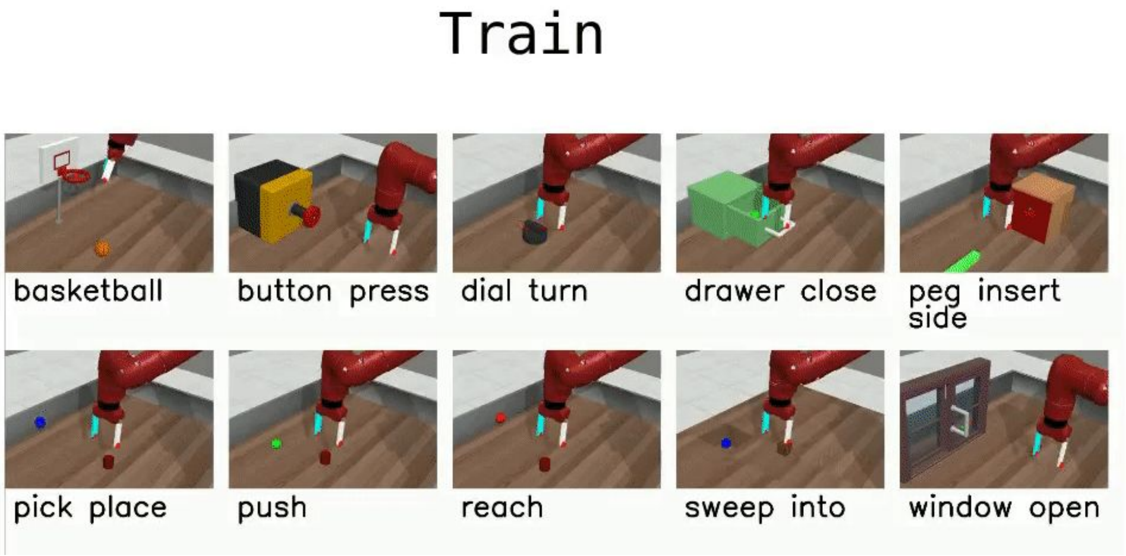
Connection to LLMs

Google

Empirical Performances

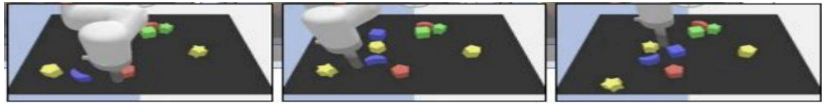
	HalfCheetah	Humanoid	Walker	Ant	Hopper
μLV-Rep	**3596.2 ± 874.5**	**806.7 ± 120.7**	**1298.1± 276.3**	**1621.4 ± 472.3**	**1096.4 ± 130.4**
Dreamer-v2	2863.8 ± 386	672.5 ± 36.6	**1305.8 ± 234.2**	1252.1 ± 284.2	758.3 ± 115.8
SAC-MLP	1612.0 ± 223	242.1 ± 43.6	736.5 ± 65.6	**1612.0 ± 223**	614.15 ± 67.6
SLAC	**3012.4** ± 724.6	387.4 ± 69.2	536.5 ± 123.2	1134.8 ± 326.2	739.3 ± 98.2
PSR	2679.75±386	534.4 ± 36.6	862.4 ± 355.3	1128.3 ± 166.6	818.8 ± 87.2
Best-FO	5557.6±439.5	1086±278.2	2523.5±333.9	2511.8±460.0	2204.8±496.0
	Cheetah-run	Walker-run	Hopper-run	Humanoid-run	Pendulum
μLV-Rep	525.3 ± 89.2	**702.3 ± 124.3**	**69.3± 12.8**	**9.8 ± 6.4**	**168.2 ± 5.3**
Dreamer-v2	**602.3 ± 48.5**	438.2 ± 78.2	**59.2 ± 15.9**	2.3 ± 0.4	**172.3 ± 8.0**
SAC-MLP	483.3 ± 77.2	279.8 ± 190.6	19.2 ± 2.3	1.2 ± 0.1	163.6 ± 9.3
SLAC	105.1 ± 30.1	139.2 ± 3.4	36.1 ± 15.3	0.9 ± 0.1	**167.3 ± 11.2**
PSR	173.7 ± 25.7	57.4 ± 7.4	23.2 ± 9.5	0.8 ± 0.1	159.4 ± 9.2
Best-FO	639.3±24.5	724.2±37.8	72.9±40.6	11.8±6.8	167.1±3.1

Video-based Reinforcement Learning



Google

Foundation Models for Representation Learning

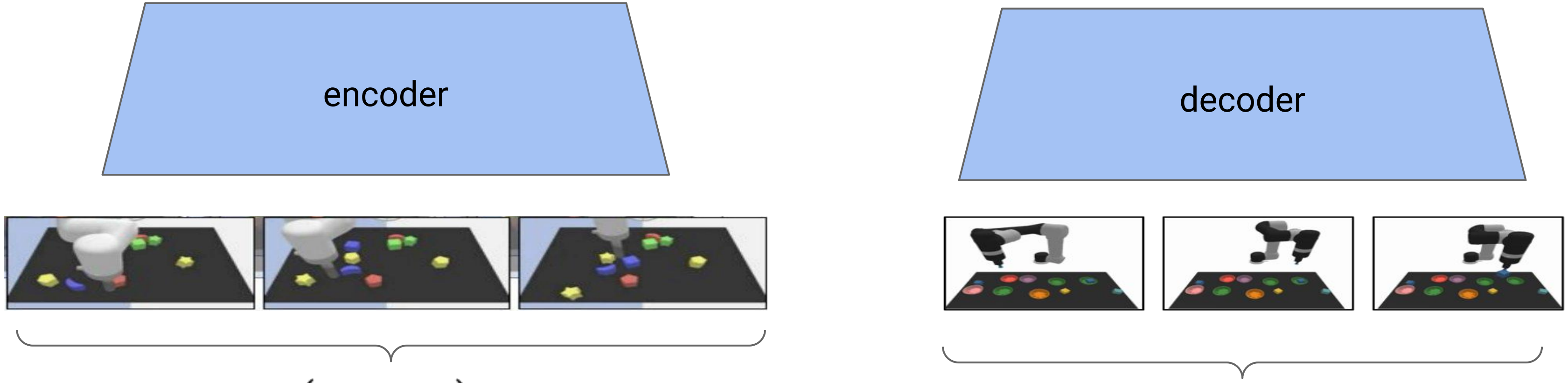


(x_h, a_h)

$(o_{h+1}, o_{h+2}, o_{h+3})$

Google

Foundation Models for Representation Learning

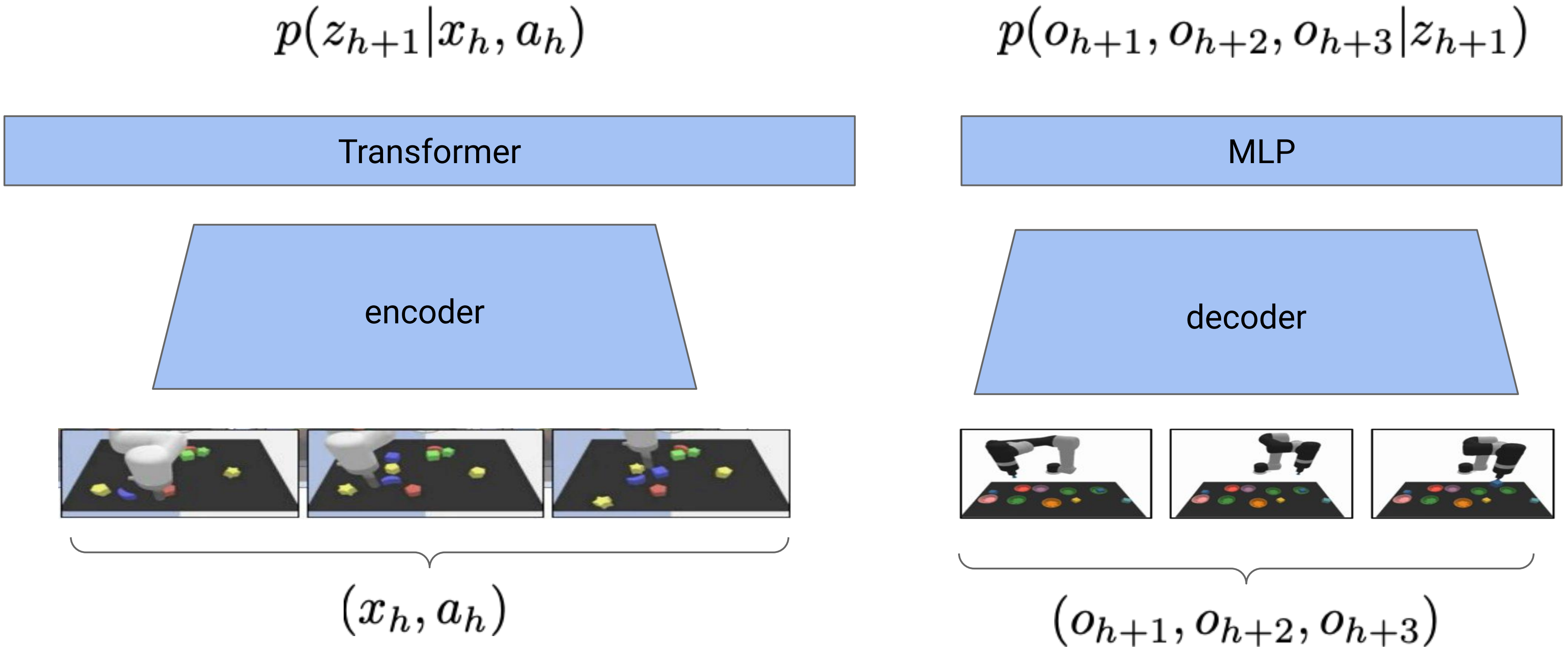


(x_h, a_h)

$(o_{h+1}, o_{h+2}, o_{h+3})$

Google

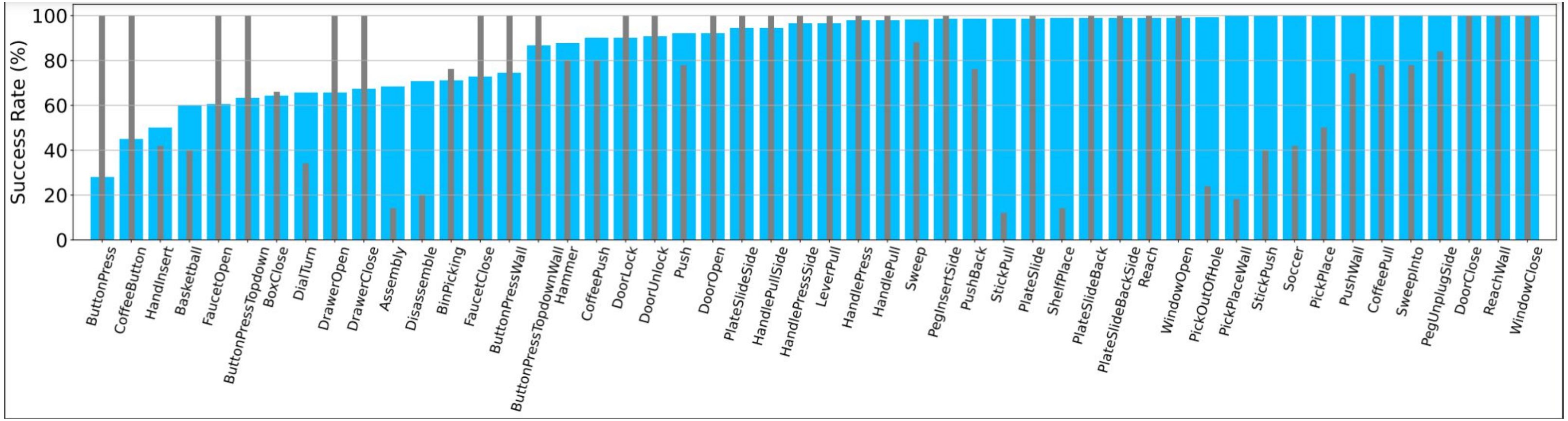
Foundation Models for Representation Learning



Linear Representation for POMDPs

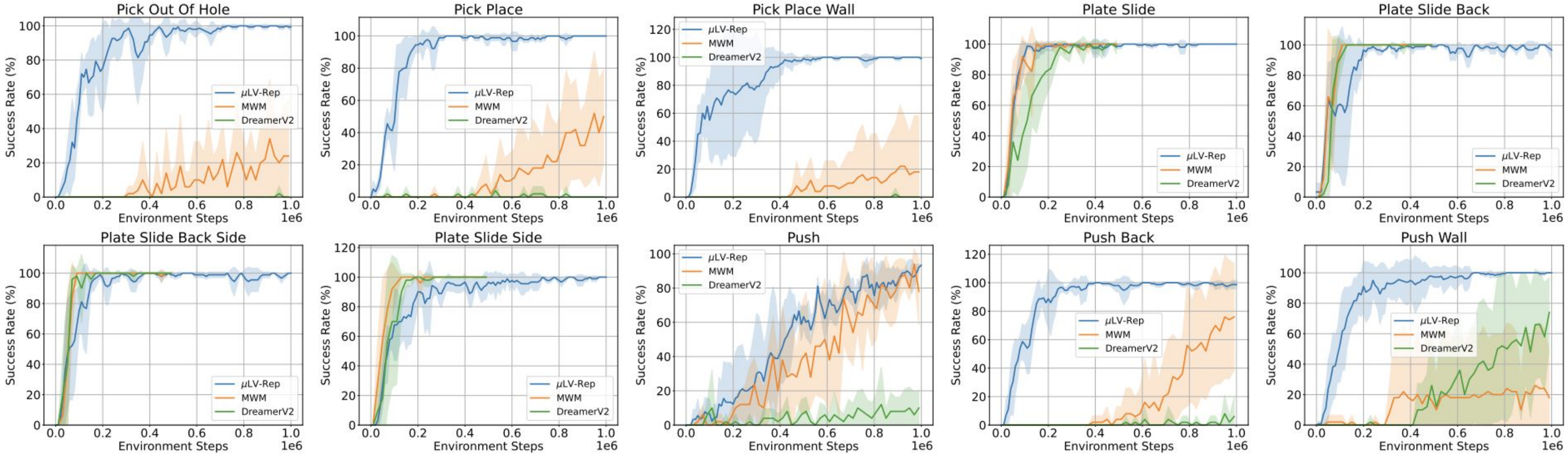
$$
\begin{aligned}
&\log p\left(o_{h+1:h+l} | x_h, a_h\right) = \log \int_{\mathcal{Z}} p(z_h | x_h, a_h) \mathbb{P}^{\pi}\left(o_{h+1:h+l} | z_h\right) \\
&= \log \int_{\mathcal{Z}} \frac{p(z_h | x_h, a_h) \mathbb{P}^{\pi}\left(o_{h+1:h+l} | z_h\right)}{q(z | x_h, a_h, o_{h+1:h+l})} q(z | x_h, a_h, o_{h+1:h+l}) \\
&= \max_{q \in \Delta(\mathcal{Z})} \mathbb{E}_{q(\cdot | x_h, a_h, o_{h+1:h+l})}\left[\log \mathbb{P}^{\pi}\left(o_{h+1:h+l} | z_h\right)\right] - D_{KL}\left(q(z | x_h, a_h, o_{h+1:h+l}) || p(z_h | x_h)\right),
\end{aligned} \tag{17}
$$

Empirical Performances



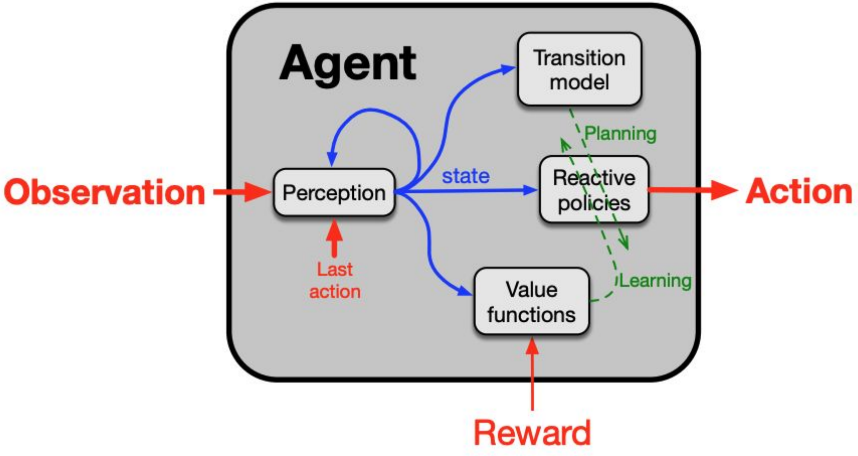
Google

Empirical Performances

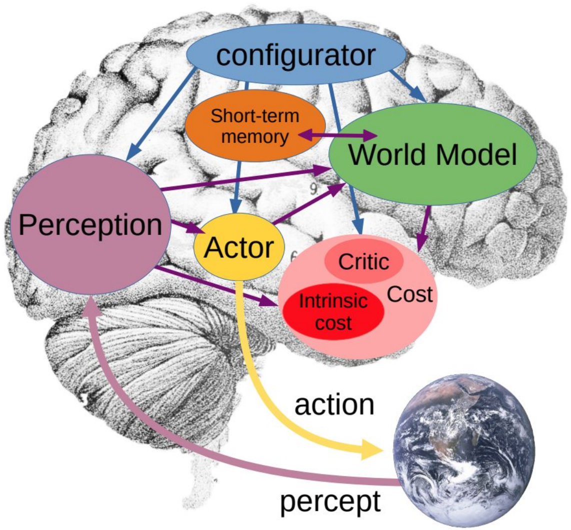


Google

Positioning in the Big Picture



Rich Sutton, 2022



Yann LeCun, 2022

References

- Tongzheng Ren*, Tianjun Zhang*, Csaba Szepesvári, Bo Dai. A Free Lunch from the Noise: Provable and Practical Exploration for Representation Learning. Conference on Uncertainty in Artifical Intelligence (**UAI**) 2022.
- Tianjun Zhang*, Tongzheng Ren*, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, Bo Dai. Making Linear MDP Practical via Contrastive Representation Learning. International Conference on Machine Learning (**ICML**) 2022.
- Tongzheng Ren*, Tianjun Zhang*, Lisa Lee, Joseph Gonzalez, Dale Schuurmans, Bo Dai. Spectral Decomposition Representation for Reinforcement Learning. International Conference on Learning Representations (**ICLR**) 2023.
- Tongzheng Ren, Chenjun Xiao, Tianjun Zhang, Na Li, Zhaoran Wang, Sujay Sanghavi, Dale Schuurmans, Bo Dai. Latent Variable Representation for Reinforcement Learning. International Conference on Learning Representations (**ICLR**) 2023.
- Tongzheng Ren, Zhaolin Ren, Na Li, Bo Dai. Stochastic Nonlinear Control via Finite-dimensional Spectral Dynamic Embedding. IEEE Conference on Decision and Control (**CDC**) 2023.
- Hongming Zhang, Tongzheng Ren, Chenjun Xiao, Dale Schuurmans, Bo Dai. Provable Representation with Efficient Planning for Partially Observable Reinforcement Learning. ArXiv 2023.
- Yang Hu, Tianyi Chen, Na Li, Kai Wang, Bo Dai. Primal-Dual Spectral Representation for Off-policy Evaluation. ArXiv, 2024

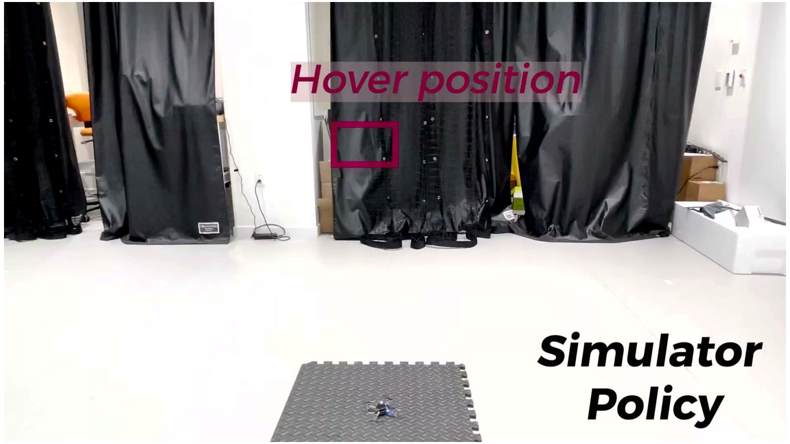
Google

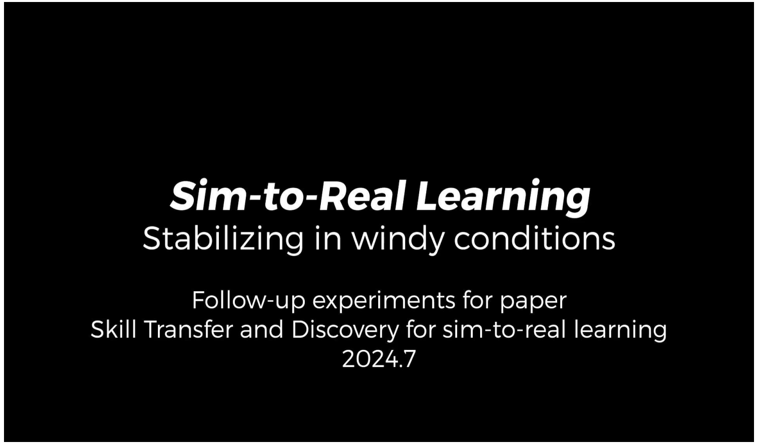
More Recent Progress

- Dmitry Shribak, Chen-Xiao Gao, Yitong Li, Chenjun Xiao, Bo Dai. Diffusion Spectral Representation for Reinforcement Learning. NeurIPS, 2024.
- Haitong Ma, Zhaolin Ren, Bo Dai, Na Li. Skill Transfer and Discovery for Sim-to-Real Learning: A Representation-Based Viewpoint. IROS, 2024
- Zhaolin Ren, Runyu Zhang, Bo Dai, Na Li. Scalable Spectral Representations for Network Multi-Agent Control. ArXiv, 2024
- Haitong Ma, Bo Dai, Zhaolin Ren, Yebin Wang, Na Li. Offline Imitation Learning upon Sub-optimal Demonstrations by Primal-Dual Representation. Submitted.

Sim-to-Real

- Haitong Ma, Zhaolin Ren, Bo Dai, Na Li. Skill Transfer and Discovery for Sim-to-Real Learning: A Representation-Based Viewpoint. IROS, 2024





Code

Spectral Representation for RL: https://github.com/haotiansun14/rl-rep

Sim-to-Real: https://congharvard.github.io/steady-sim-to-real/

Thanks!
Questions?

Spectral View of Representations

Representation	Decomposed Dynamics
Successor Feature	$\texttt{svd}\left((I-\gamma P^{\pi})^{-1}\right)$
Proto-Value Function	$\texttt{eig}\left(\Lambda P^{\pi} + (P^{\pi})^{\top}\Lambda\right)$
Krylov Basis	$\left\{(P^{\pi})^{i} r\right\}_{i=1}^{k}$
Spectral State-Aggregation	$\texttt{svd}\left(P^{\pi}\right)$

Markov Decision Processes (MDPs)

Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, T, \mu, H \rangle$

- State space: $\mathcal{S}$
- Action space: $\mathcal{A}$
- Reward function: $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition: $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
- Initial state distribution: μ

LQR can be reformulated as a special case of MDP

NCE approaches MLE

$$\lim_{K\to\infty}\sum_{j=1}^{K}\log(1-h(y_{i,j},u_i;\gamma)) = -\lim_{K\to\infty}\frac{\sum_{j=1}^{K}\log\left(1+\frac{f(y_{i,j},u_i)\exp(-\gamma)}{Kq(y_{i,j})}\right)^{K}}{K}$$

$$= -\lim_{K\to\infty}\frac{\sum_{j=1}^{K}\left(\frac{f(y_{i,j},u_i)\exp(-\gamma)}{q(y_{i,j})}\right)}{K} = -\mathbb{E}_{y_i\sim q(y)}\left[\frac{f(y_i,u_i)\exp(-\gamma)}{q(y_i)}\right] = -\exp(-\gamma)\int f(y,u_i)dy.$$

NCE approaches MLE

$$\lim_{K\to\infty}[\log(h(x_i,u_i;\gamma))+\log K]=\log\frac{f(x_i,u_i)}{q(x_i)}-\gamma-\lim_{K\to\infty}\log\left(1+\frac{f(y_i,u_i)\exp(-\gamma)}{Kq(y_i)}\right)$$
$$=\log\frac{f(x_i,u_i)}{q(x_i)}-\gamma.$$

Hence,

$$\lim_{K\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[\log(h(x_i,u_i;\gamma))+\sum_{j=1}^{K}\log(1-h(y_{i,j},u_i;\gamma))\right]+\log K=\frac{1}{n}\sum_{i=1}^{n}\log\frac{f(x_i,u_i)}{q(x_i)}-\gamma-\exp(-\gamma)\int f(y,u_i)dy.$$