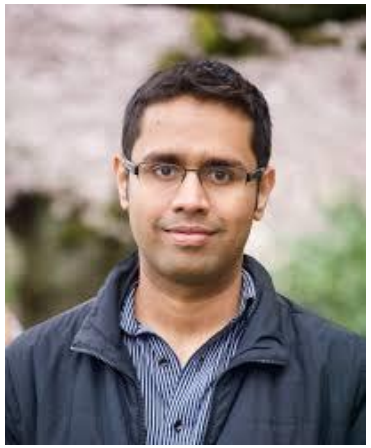
MOReL: Model-Based Offline Reinforcement Learning



Rahul Kidambi

Cornell University



Aravind Rajeswaran

University of Washington



Praneeth Netrapalli

Microsoft Research India



Thorsten Joachims

Cornell University

(Deep) Reinforcement Learning

Many application domains

- Robotics/Bio-mechanics

[Lilicrap et al `16, OpenAI `18]

- Ranking/Recommender Systems

[Covington et al. `18, Chen et al. `18]

- Games

[Mnih et al `15, Silver et al 2016]

- Graphics (The Incredibles!!)

[Mordatch et al `15, Peng et al `18]

Credit: Youtube

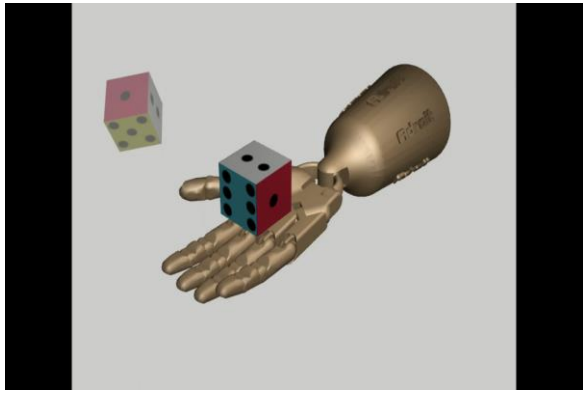


Credit: Disney



ATARI
GAMES

Credit: atari.com



Credit: Rajeswaran et al. (2018)

MDP: Preliminaries

$$\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, T)$$

State - $\mathcal{S} \subset \mathrm{R}^d$ - grid position

Action - $\mathcal{A} \subset R^{d'}$ (up, down, left, right)

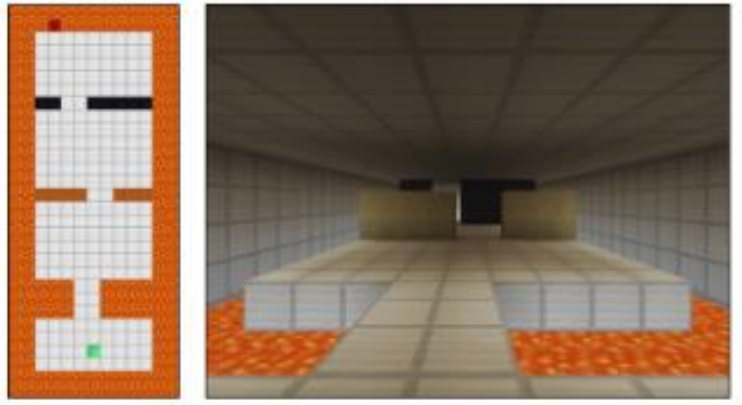
Transition - $\mathcal{T}: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ – next grid position

Reward $R: \mathcal{S} \times \mathcal{A} \to \mathrm{R}$ – Distance to target

Start state distribution $\rho_0 \in \mathcal{S}$ – where you begin.

Policy – $\pi: \mathcal{S} \to \mathcal{A}$

Image courtesy: Johnson et al. (2016)



$\mathrm{Interact}(\pi, \mathcal{M})$:

- $s_0 \sim \rho_0$ (start state)
- For $\mathrm{t} = 0,1,\cdots,T-1$:
  - $a_t \sim \pi(\cdot\,|s_t)$ (movement)
  - $r_t \leftarrow r(s_t, a_t)$
  - $s_{t+1} \sim \mathcal{T}(\cdot\,|s_t, a_t)$ (dynamics)

Return $\mathrm{Traj} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$

Goal: Compute

$$\pi^* = \mathrm{argmax}_{\pi \in \Pi} V(\pi) = \mathrm{E}[r_0 + \gamma \cdot r_1 + \cdots \gamma^{T-1} \cdot r_{T-1} | \pi]$$

Approaches to solve RL



Model

Model Based

Value Based

Actor Critic

Policy Based

Value

Policy

Model Free



(Generalized) Policy Iteration
Policy Gradient + Critic (value)
(Asynchronous) Advantage Actor-Critic (A2C/A3C)
Soft Actor-Critic (SAC)

(Fitted) Value Iteration
SARSA
(Deep) Q-learning

Policy Gradient
NPG/TRPO/PPO etc.
Zeroth order optimization

Adapted from David Silver's slides

(Online) Policy Gradient

PG [Sutton et al. 2000], NPG [Kakade `01], TRPO [Schulman et al. `15], PPO [Schulman et al. `17]

$\text{Plan}(\pi_0, \mathcal{M}, K)$:

$\pi \leftarrow \pi_0$

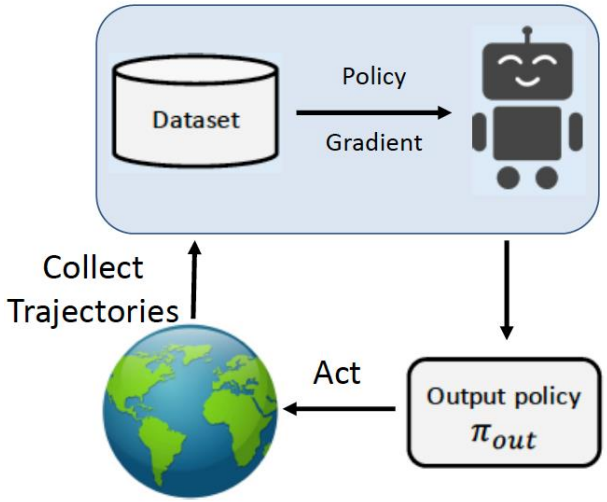
REPEAT Till CONVERGENCE:

$\text{Paths} \leftarrow \text{Interact}(\pi, \mathcal{M})\ \ K - \text{times}$

$\nabla V\ (\pi) = \text{GetPolicyGradient}(\text{Paths})$

$\pi \leftarrow \pi + \gamma \cdot \nabla V\ (\pi)$

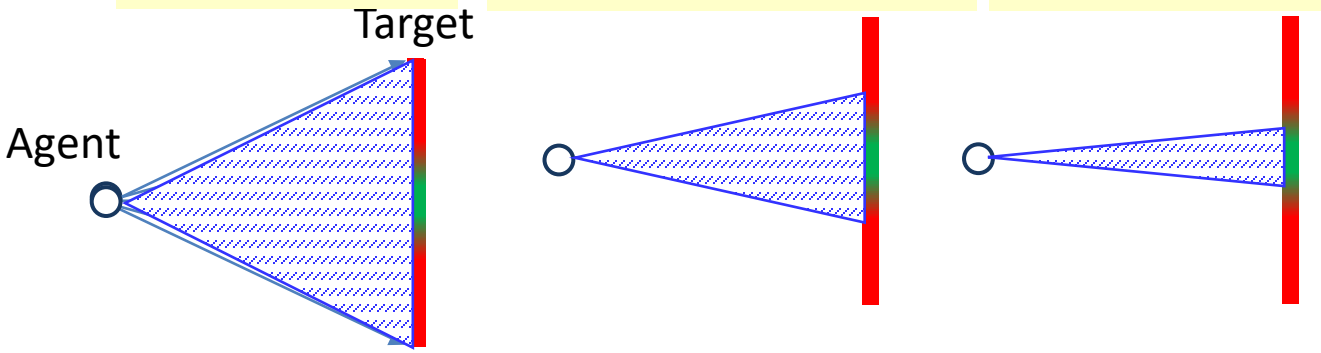
Return π



Positives:
-- Monotonic progress (in theory)
-- Global optimality [Agarwal et al. `19]

Issues:
-- Sample efficiency
-- Non-monotonicity in practice
(baselines, func. approx., noisy gradients)

Random Policy | Partially Trained Policy | Optimal Policy



Broad state visitation distribution

States visited by the policy

Credit: unsplash.com

(Online) Policy Gradient

PG [Sutton et al. 2000], NPG [Kakade `01], TRPO [Schulman et al. `15], PPO [Schulman et al. `17]

$\text{Plan}(\pi_0, \mathcal{M}, K)$:

$\pi \leftarrow \pi_0$

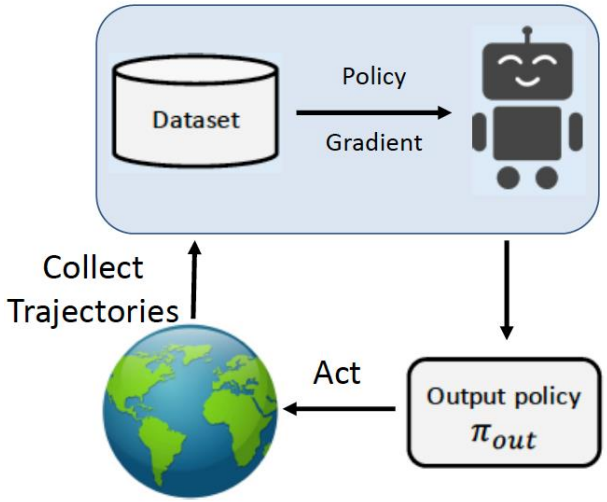
REPEAT Till CONVERGENCE:

$\text{Paths} \leftarrow \text{Interact}(\pi, \mathcal{M})\ \ K - \text{times}$

$\nabla V\ (\pi) = \text{GetPolicyGradient}(\text{Paths})$

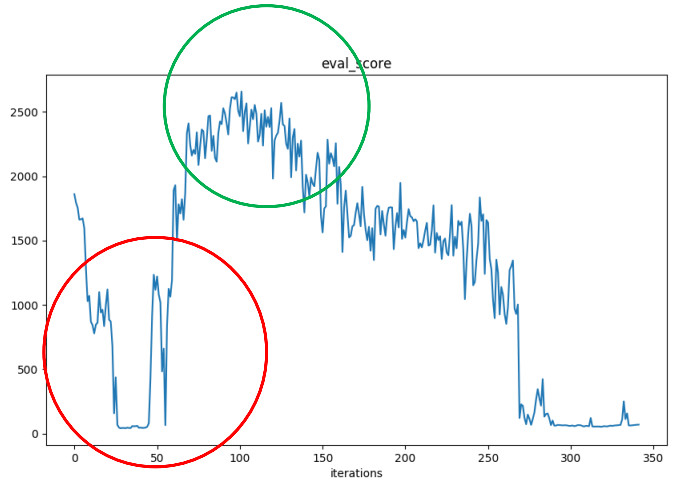
$\pi \leftarrow \pi + \gamma \cdot \nabla V\ (\pi)$

Return π



Positives:
-- Monotonic progress (in theory)
-- Global optimality [Agarwal et al. `19]

Issues:
-- Sample efficiency
-- Non-monotonicity in practice
(baselines, func. approx., noisy gradients)





Credit: unsplash.com

Offline Reinforcement Learning

$\text{OfflineRL}(\pi_b, \text{N}, \mathcal{M})$:

For $\text{i} = 1 \cdots \text{N}$:

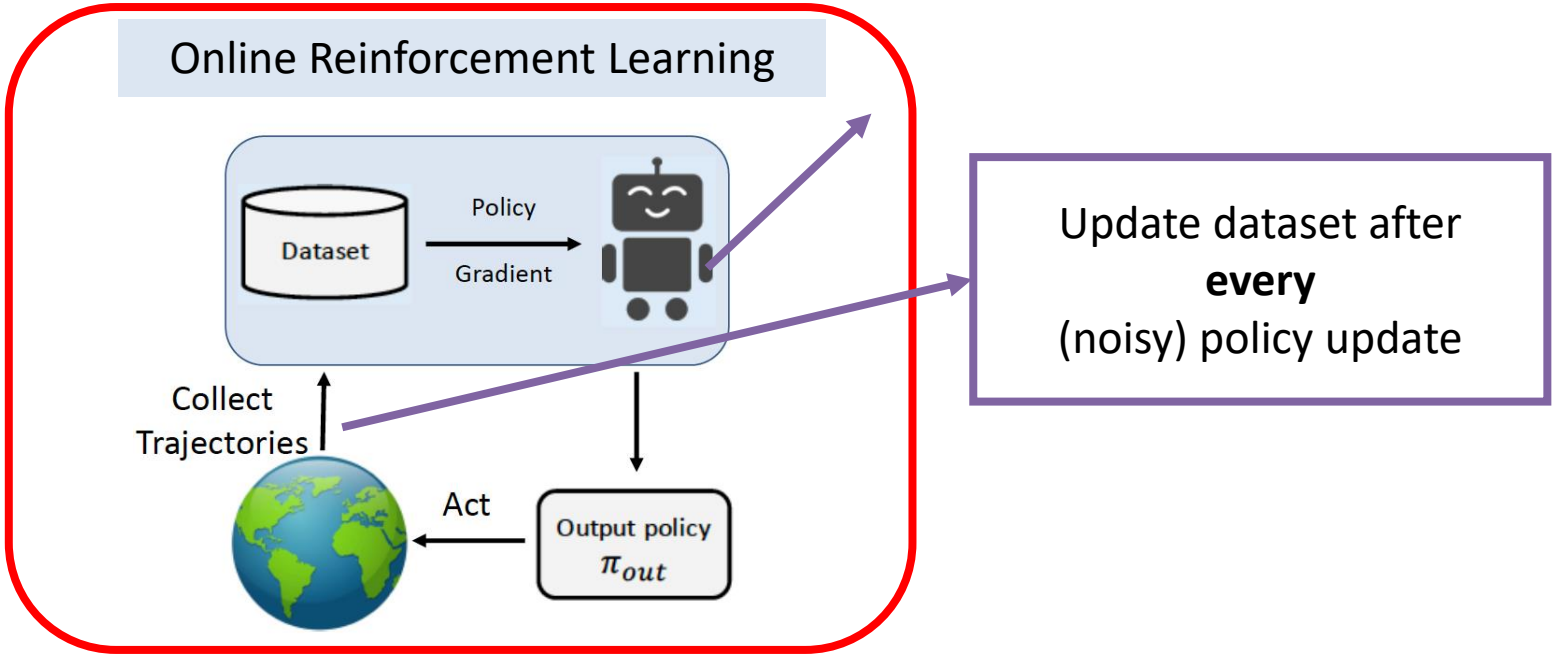
$\text{Traj[i]} = \text{Interact}(\pi_b, \mathcal{M})$

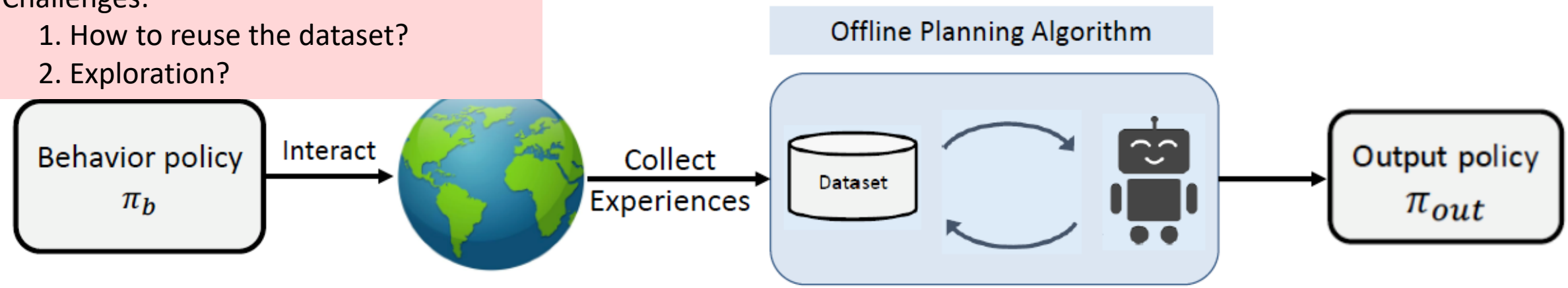
$\pi_{\text{out}} \leftarrow \text{OfflinePlanningAlgorithm}(\text{Traj})$

Return π_{out}

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. How to reuse the dataset?
2. Exploration?





Offline Reinforcement Learning

$\text{OfflineRL}(\pi_b, \text{N}, \mathcal{M})$:

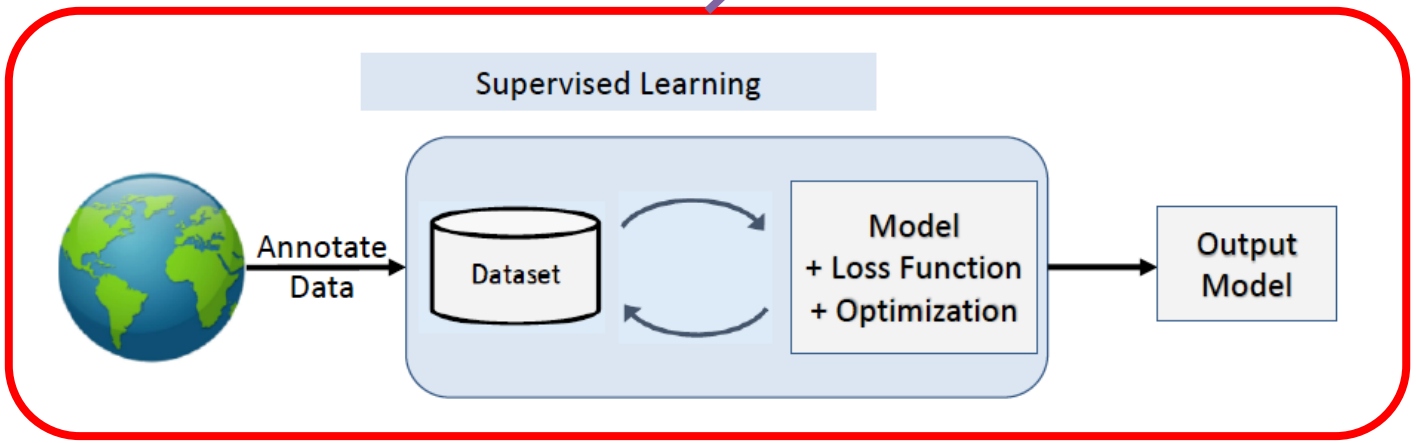
For $\text{i} = 1 \cdots \text{N}$:

$\text{Traj[i]} = \text{Interact}(\pi_b, \mathcal{M})$

$\pi_{\text{out}} \leftarrow \text{OfflinePlanningAlgorithm}(\text{Traj})$

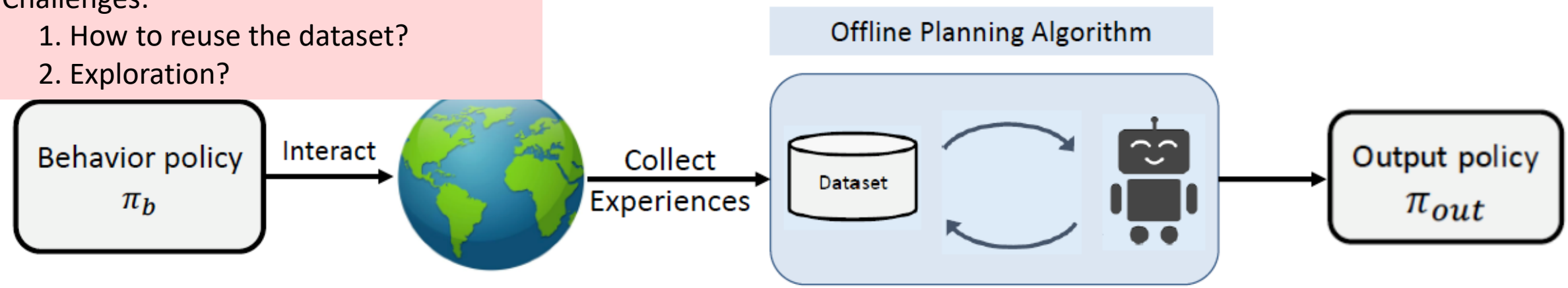
Return π_{out}

Offline RL - **strictly harder**
1. Sequential Decision Making
2. Partial (Bandit) Feedback



Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. How to reuse the dataset?
2. Exploration?



Challenges In Offline RL

$\text{OfflineRL}(\pi_b, \text{N}, \mathcal{M})$:

For $\text{i} = 1 \cdots \text{N}$:

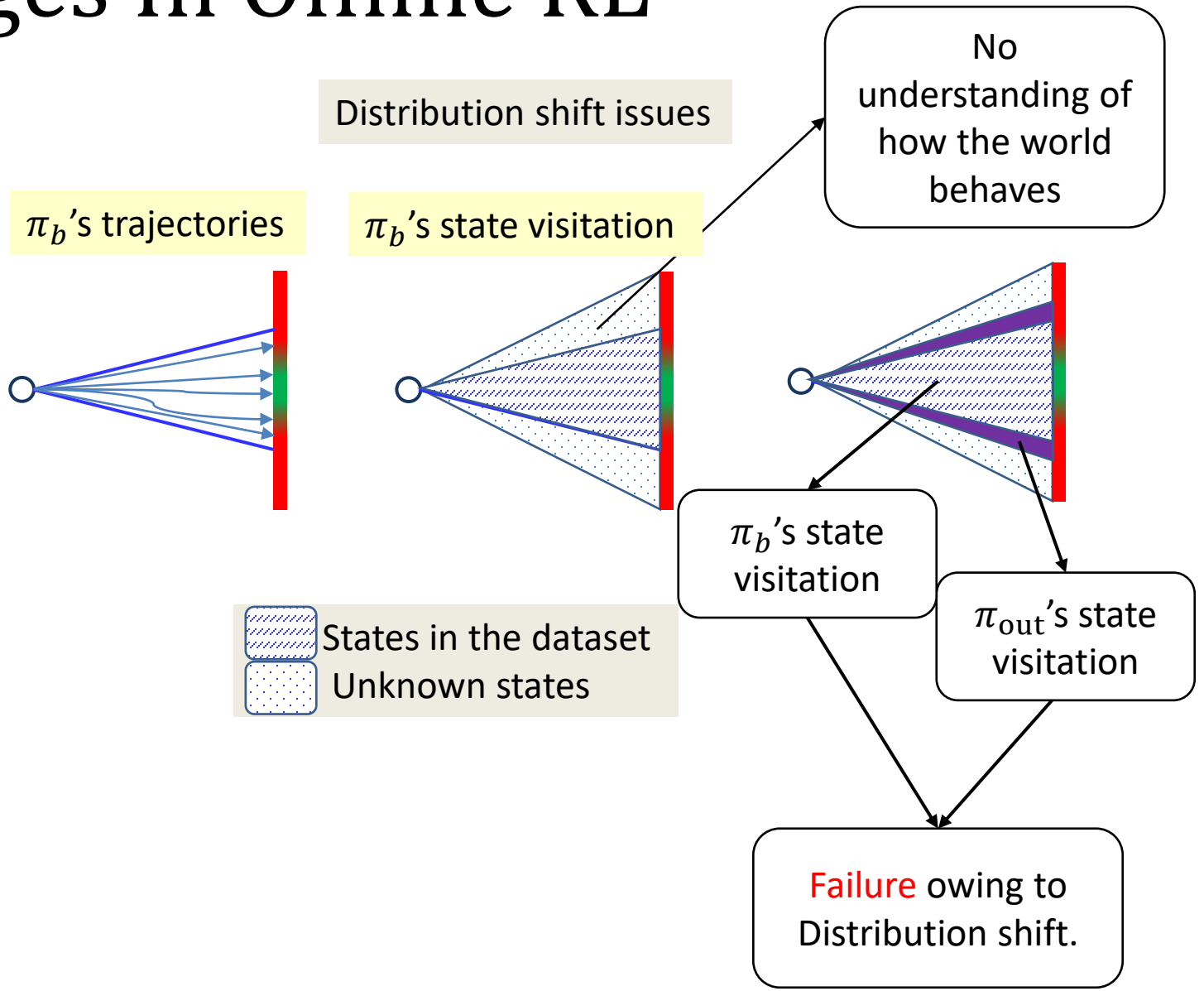
$\text{Traj[i]} = \text{Interact}(\pi_b, \mathcal{M})$

$\pi_{\text{out}} \leftarrow \text{OfflinePlanningAlgorithm(Traj)}$

Return π_{out}

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. Exploration?
2. How to reuse the dataset?
3. **Distribution Shift**
4. Impact of the logging policy?



Challenges In Offline RL

$\text{OfflineRL}(\pi_b, \text{N}, \mathcal{M})$:

For $\text{i} = 1 \cdots \text{N}$:

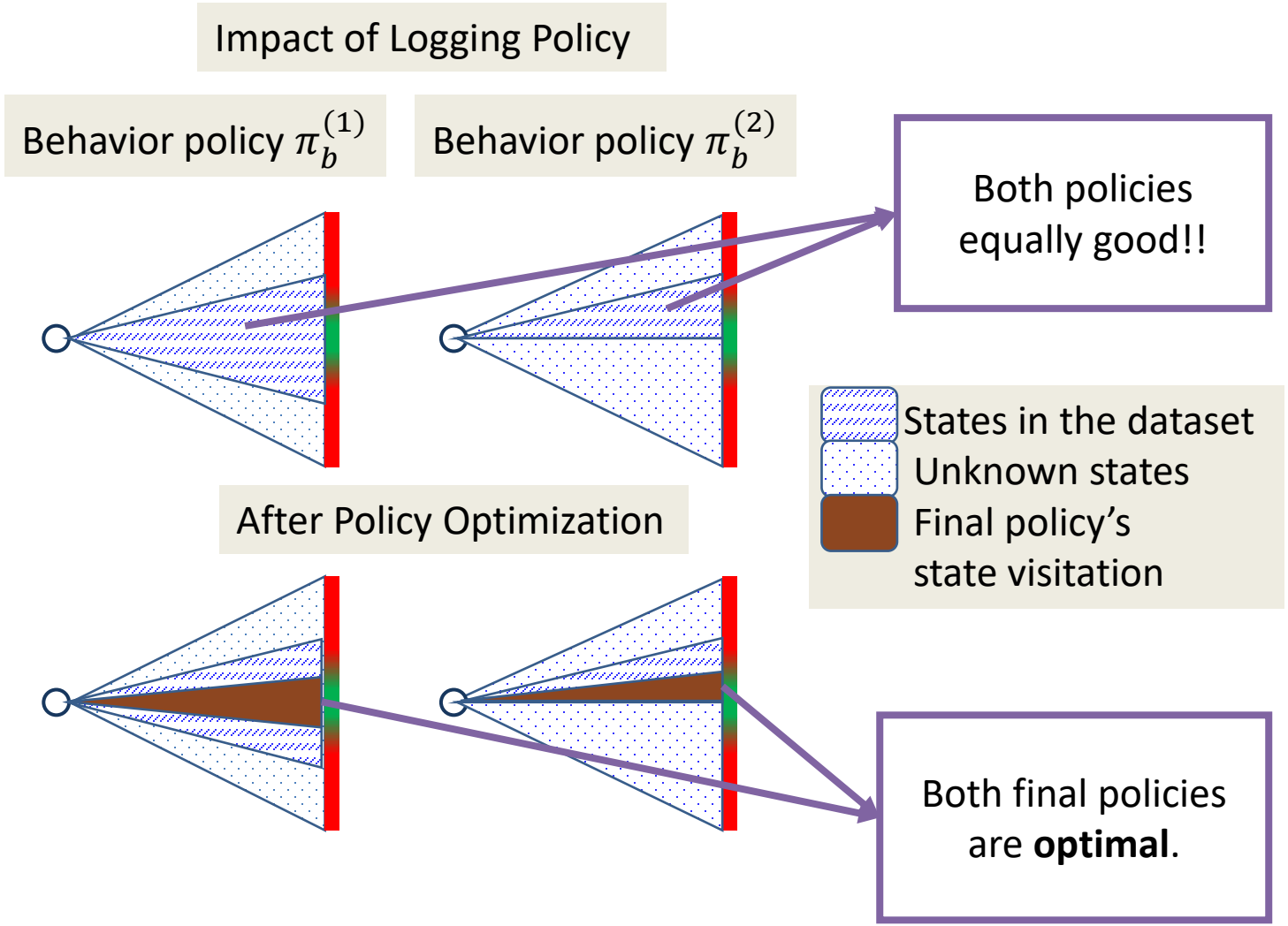
$\text{Traj[i]} = \text{Interact}(\pi_b, \mathcal{M})$

$\pi_{\text{out}} \leftarrow \text{OfflinePlanningAlgorithm(Traj)}$

Return π_{out}

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. Exploration?
2. How to reuse the dataset?
3. Handling distribution shift
4. **Impact of the logging policy?**



Prior Work

Direction 1: Off-Policy Actor-Critic

- Q-learning [Watkins `89]
  - Good performance in tabular settings [Watkins & Dayan 92]
  - Challenging with function approx. [Baird ‘95, Sutton ‘95]
    - Setting targets for Bellman error minimization
    - Replay buffer dictates performance.

- Actor-Critic methods
  - DDPG [Silver et al`14], DQN [Mnih et al`15], SAC [Haarnoja et al `18]:
  - Similar issues [van Hasselt et al`18, Fujimoto et al`18, Fu et al`19]

- Many recent works [Fujimoto et al`18b, Kumar et al`19, Jaques et al`19, Wu et al`19]
  - Design choices vary on a per-task basis.

Direction 2: Off-policy Policy Gradients

- Use importance weighted policy gradient
  - Importance weights: exponential in horizon.
    - Fix: use state-distribution ratios [Liu et al.`18].
  - Maintain ESS is reasonably large [Fakoor et al. `18].

- Liu et al. (2018)
  - Model-free planning using pessimistic MDP approaches.
  - Yet to be examined for continuous control tasks.

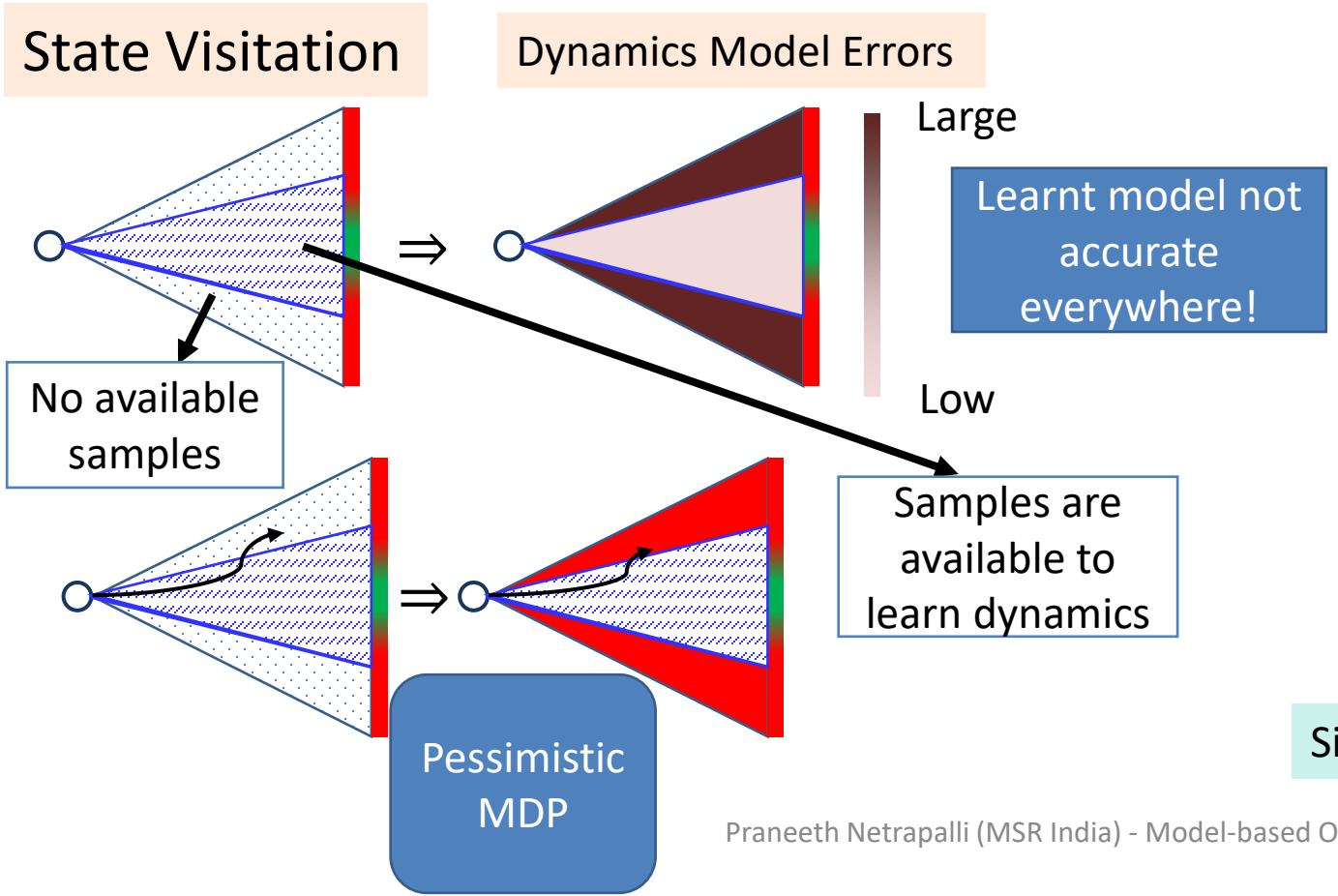
- Many such issues examined in contextual bandit problems
  - For e.g. [Swaminathan & Joachims `15, Joachims et al. `18, Ma et al. `19] and many others.

Our Framework: MOReL

MOReL is an offline RL framework: relies on optimization + supervised learning.

One-line description: plan using a model-based pessimistic MDP construction.

Achieves state-of-the-art on several benchmarks.



Features of MOReL

Model-based approach:
- Effective sample reuse
- Unlocks generalization!
- On-policy samples

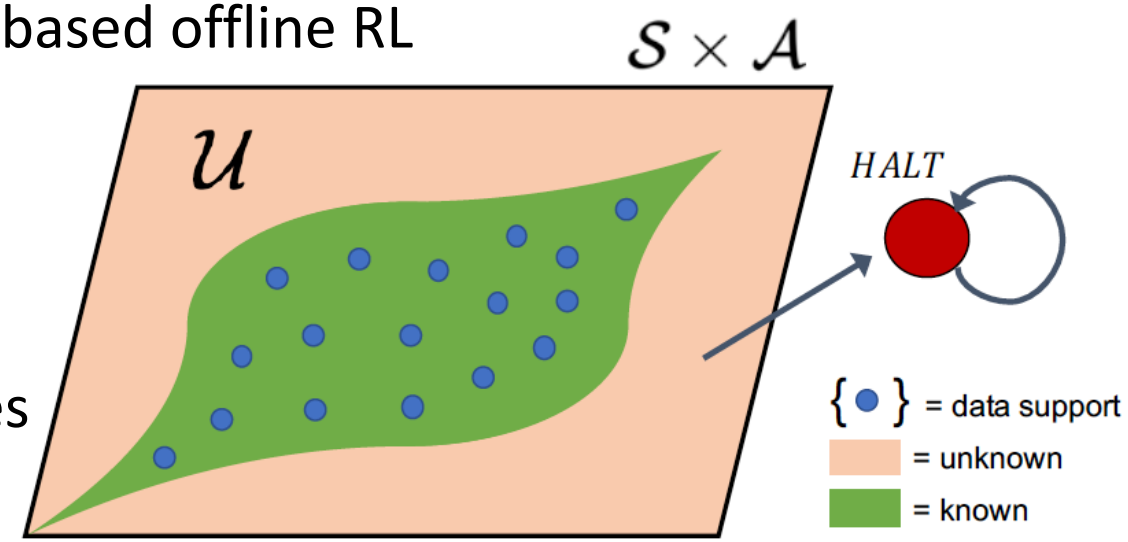
Caveat: No model is globally accurate

Planning based on a "pessimistic" MDP framework:
- Effective way to handle distribution shift.
- Use model where it is accurate!

Similar algorithm proposed independently by Yu et al. `20

Our Contributions

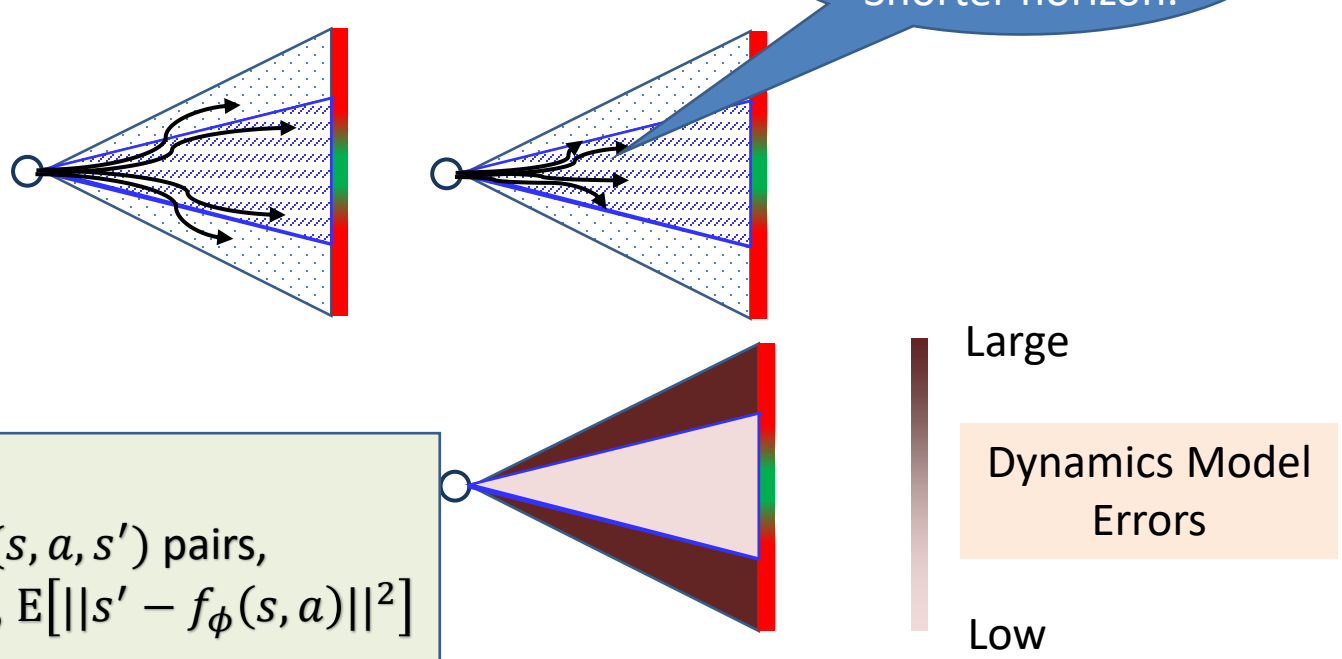
- MOReL: model-based offline RL
  - Ross and Bagnell (2012) analyzed naïve model-based offline RL
- Pessimistic MDP construction
  - State-action pairs → known/unknown
- Planning on the pessimistic MDP
  - Policy discouraged from visiting unknown states
- MOReL - minimax optimal for offline RL
  - Model score approx. lower bounds true score
- Experiments on deep RL benchmarks
  - Continuous state/action spaces with function approx.



Review: Model-Based RL

- Replace $\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, T_h)$
  with $\widehat{\mathcal{M}} \equiv (\mathcal{S}, \mathcal{A}, \hat{\mathcal{T}}, R, \rho_0, T_h)$,
- Where, $\hat{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is estimated through MLE on the logged dataset.
- Plan using $\widehat{\mathcal{M}}$.

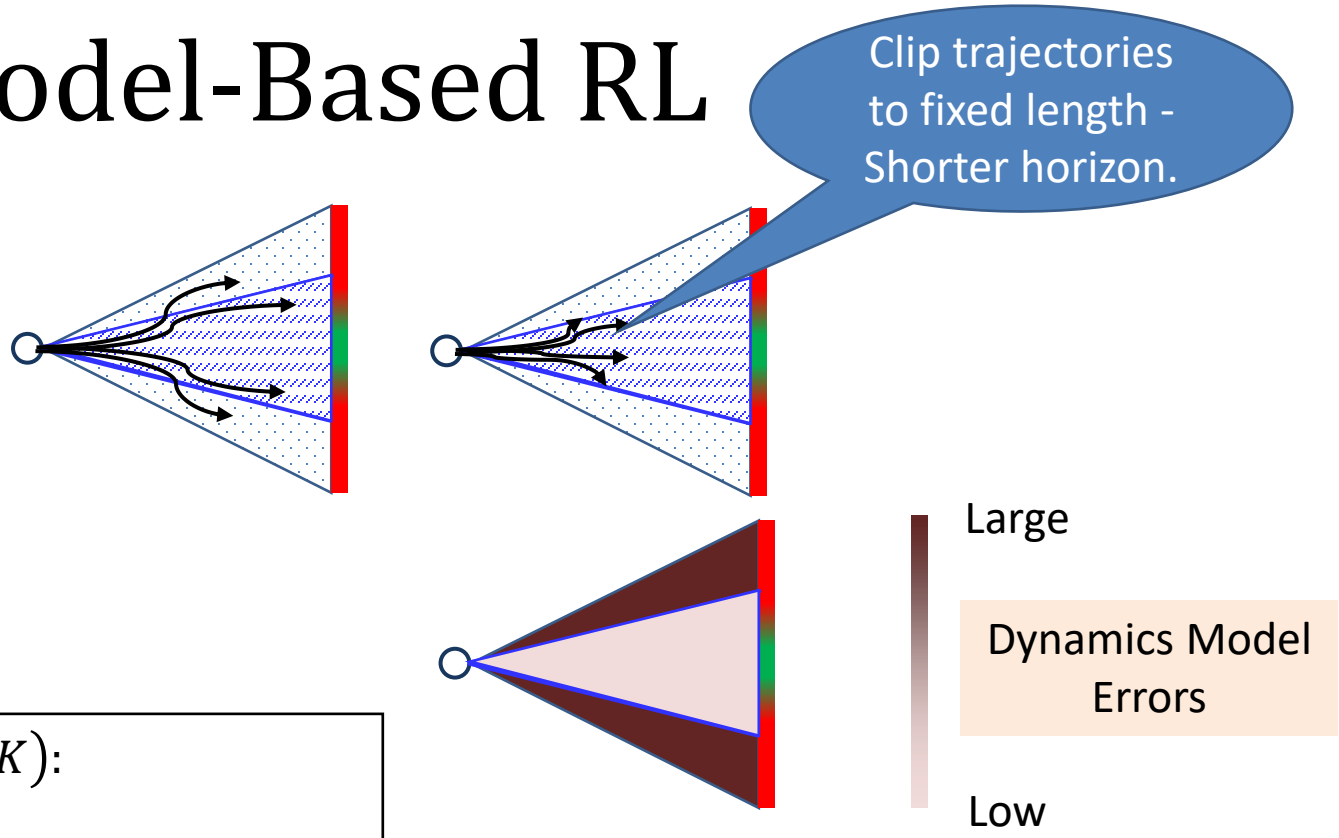
Given (s, a, s') pairs,
$\hat{\mathcal{T}}_\phi = \text{argmin}_\phi \, \mathrm{E}[||s' - f_\phi(s, a)||^2]$



Advantages:
- Sample efficiency & reuse.

Review: Model-Based RL

- Replace $\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, T_h)$ with $\widehat{\mathcal{M}} \equiv (\mathcal{S}, \mathcal{A}, \hat{\mathcal{T}}, R, \rho_0, T_h)$,
- Where, $\hat{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is estimated through MLE on the logged dataset.
- Plan using $\widehat{\mathcal{M}}$.



Advantages:
- Sample efficiency & reuse.

$\text{Plan}(\pi_0, \widehat{\mathcal{M}}, K)$:

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

$\text{Paths} \leftarrow \text{Interact}(\pi, \widehat{\mathcal{M}})$ $K-$ times

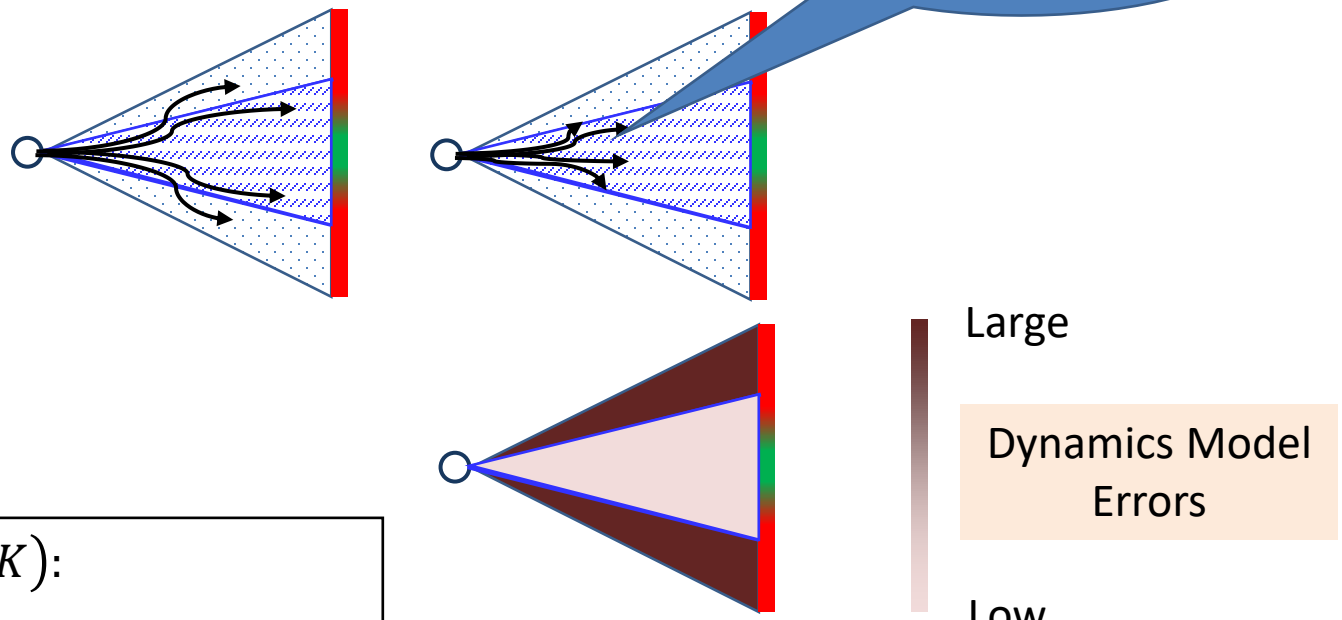
$\text{PG}(\pi) = \text{GetPolicyGradient}(\text{Paths})$

$\pi \leftarrow \pi_n + \gamma \cdot \text{PG}(\pi)$

Return π

Review: Model-Based RL

- Replace $\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, T_h)$ with $\widehat{\mathcal{M}} \equiv (\mathcal{S}, \mathcal{A}, \hat{\mathcal{T}}, R, \rho_0, T_h)$,
- Where, $\hat{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is estimated through MLE on the logged dataset.
- Plan using $\widehat{\mathcal{M}}$.



$\text{Plan}(\pi_0, \widehat{\mathcal{M}}, K)$:

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

$\text{Paths} \leftarrow \text{Interact}(\pi, \widehat{\mathcal{M}})$ $K-$ times

Return π

Advantages:
- Sample efficiency & reuse.

Issues:
- short horizon bias (compounding errors) [Levine et al. 2020]
- continuous state space ⇒ model not accurate everywhere..

Suppose $|\mathcal{T} - \hat{\mathcal{T}}| < \alpha$ in t.v.,
Trajectory length $\approx O(1/\alpha)$

Pessimistic MDP Formulation

Instead of $\widehat{\mathcal{M}} \equiv \left(\mathcal{S}, \mathcal{A}, \hat{\mathcal{T}}, R, \rho_0, T\right)$ consider

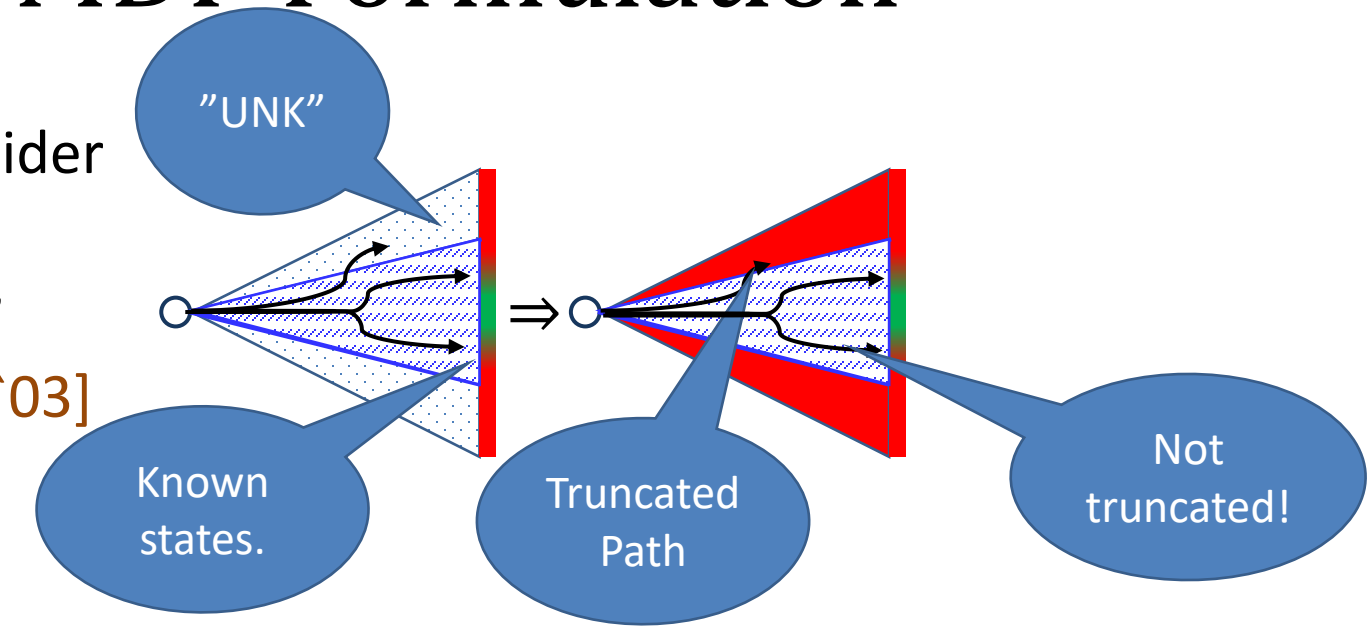
$$\widehat{\mathcal{M}_p} \equiv \left(\mathcal{S}_k \text{ U } \{\text{UNK}\}, \mathcal{A}, \hat{\mathcal{T}}, R, \rho_0, T\right),$$

- $\mathcal{S}_k$ -- set of **known** states [Kakade et al. `03]

Defining the UNK state

Ones that aren't observed in the dataset.

UNK: absorbing with large negative reward.



Advantages:

- Planner restricted to states where model is accurate.
- **Dynamic** pruning of trajectory based on **when** it encounters UNK.

Question: How do we truncate in practice?

Deciding UNK states

- Require an oracle O:

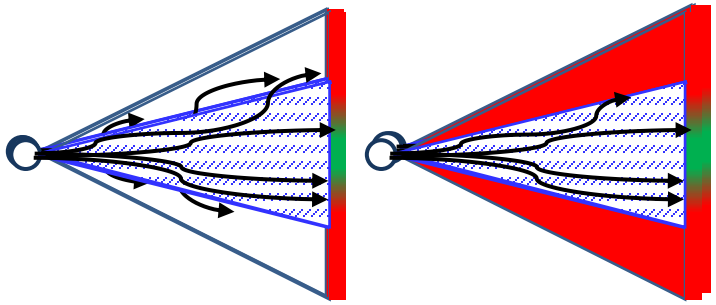
$$O(s, \text{dataset}) = \begin{cases} \text{True} & \text{if} \quad s \in \mathcal{S}_k \\ \text{False} & \text{if} \quad s \in \text{UNK} \end{cases}$$

- Examples:
  - Bayesian approaches: if $\hat{\mathcal{T}}(s, a)$ has large uncertainties, $O(s, \text{dataset}) = \text{True}$
  - (Random) Ensembles: Train multiple $\{\hat{\mathcal{T}}_i\}$.

    If these disagree on (s, a), then, $O(s, \text{dataset}) = \text{True}$

- We use the latter, but the framework is flexible.

Framework: MOReL



MOReL(Dataset, $\mathcal{M}$):

- Learn an ensemble of generative models:

$$O \equiv \{\widehat{\mathcal{T}}_i\} \leftarrow \text{MLE}(\text{Dataset}, \text{seed} = i)$$

- Define the model based pessimistic MDP:

$$\widehat{\mathcal{M}}_p = \{S_k \text{ U } \{\text{UNK}\}, \mathcal{A}, \{\widehat{\mathcal{T}}_i\}, R, \rho_0 = \text{dataset}, T_h\}$$

- $\pi_{new} \leftarrow \text{Plan}(\pi_0, \widehat{\mathcal{M}}_p, K, O)$

Plan($\pi_0, \mathcal{M}, K, O$):

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

Paths $\leftarrow$ Interact($\pi, \mathcal{M}, O$) $K -$ ti

PG (π) = GetPolicyGradient(Path

$\pi \leftarrow \pi + \gamma \cdot \text{PG}(\pi)$

Return π

Interact($\pi, \mathcal{M}, O$):

- $s_0 \sim \rho_0$ (start state)
- For $t = 0,1, \cdots, T_h - 1$:
  - If $O(s_t) = \text{True}$
    - $a_t \sim \pi(\cdot|s_t)$
    - $r_t \leftarrow r(s_t, a_t)$
    - $s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t)$
  - Else:
    - $r_t = r_{min}$
    - Break

Return Traj $= \{(s_j, a_j, r_j, s_{j+1})\}_{j=0}^{t}$

Theoretical Guarantees

Let $J_{\rho_0}(\pi, \mathcal{M})$ be the expected performance of policy π on the MDP $\mathcal{M}$, i.e.,

$$J_{\rho_0}(\pi, \mathcal{M}) = E_{s_0 \sim \rho_0, \mathcal{M}, \pi}[V(s_0)]$$

Then,

Theorem (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):

The value of any policy π the model-based pessimistic MDP $\widehat{\mathcal{M}}_{\boldsymbol{p}}$ is related to $\mathcal{M}$ as follows:

$$J_{\rho_0}\left(\pi, \widehat{\mathcal{M}}_p\right) \geq J_{\rho_0}(\pi, \mathcal{M}) - 2\,\alpha \frac{R_{\max}}{(1-\gamma)^2} - 2\frac{R_{\max}}{1-\gamma} \cdot \mathrm{E}\left[\gamma^{T_{\mathcal{U}}^{\pi}}\right]$$

$$J_{\rho_0}\left(\pi, \widehat{\mathcal{M}}_p\right) \leq J_{\rho_0}(\pi, \mathcal{M}) + 2\alpha \frac{R_{\max}}{(1-\gamma)^2}$$

Where, α is the total variation distance between the true/learnt dynamics on the known states, and $T_{\mathcal{U}}^{\pi}$ denotes the hitting time of π on unknown states in $\mathcal{M}$.

Sub-optimality bound of MOReL

Corollary (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):

Suppose the PLANNER offers an ϵ_π −sub-optimal policy on the pessimistic MDP $\widehat{\mathcal{M}}_{\boldsymbol{p}}$. Then,

$$J_{\rho_0}(\pi^*, \mathcal{M}) - J_{\rho_0}(\pi, \mathcal{M}) \leq 4 \frac{R_{\max}}{(1-\gamma)^2} \cdot \alpha - 2 \frac{R_{\max}}{(1-\gamma)} \cdot \mathrm{E}\left[\gamma^{T_{\mathcal{U}}^{\pi}}\right] + \epsilon_\pi$$

Where, α is the total variation distance between the true/learnt dynamics on the known states, and $T_{\mathcal{U}}^{\pi}$ denotes the hitting time of π on unknown states in $\mathcal{M}$.

Proposition (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):
MOReL is minimax optimal for offline RL.

Experimental Analysis

- 4 continuous control tasks from MuJoCo:
  - Hopper
  - HalfCheetah
  - Ant
  - Walker.
- Continuous state/action spaces.
- Record a total of 1M transitions.

Dataset collected with a partially-trained policy

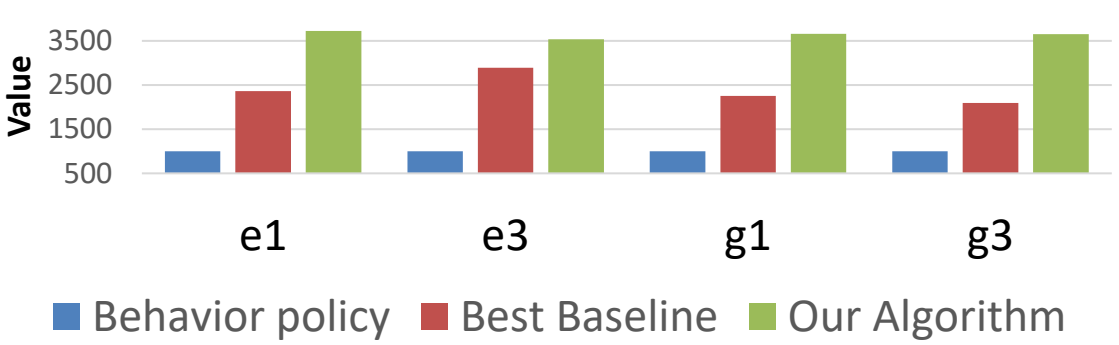
- Unroll 1000 trajectories from a sub-optimal policy.

- We compare the best of 12 recent algorithms against our approach.

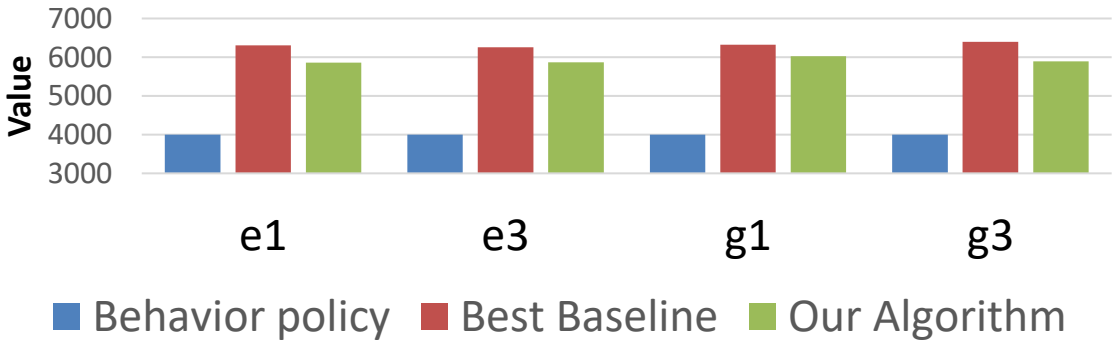
Environment	Behavior Policy Value	Best of 12 baselines	**Our Approach**
Hopper-v2	1000	2774	**3642 $\pm$ 54**
Half Cheetah-v2	4000	6207	6028 $\pm$ 192
Walker2d-v2	1000	2907	**3709 $\pm$ 159**
Ant-v2	1000	2839	**3663 $\pm$ 247**

Benchmarking with Noisy Experience Datasets

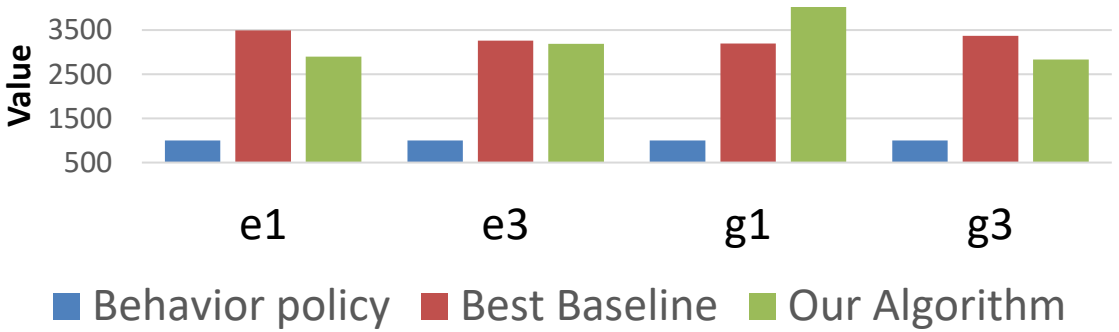
Hopper-v2



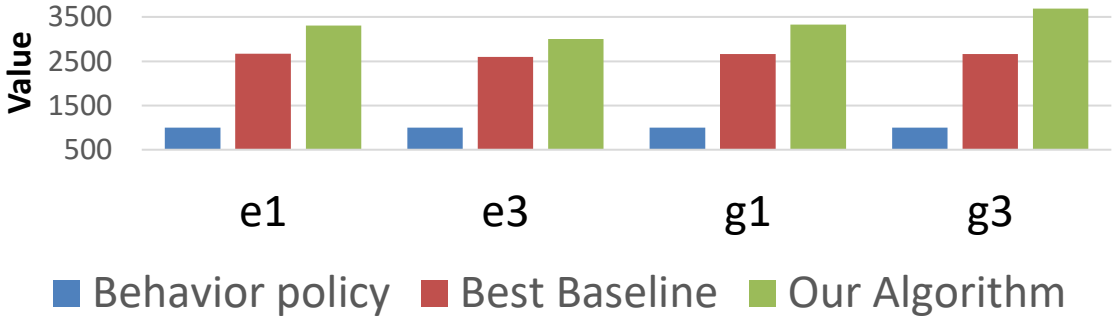
HalfCheetah-v2



Walker2d-v2



Ant-v2



Future Work

- What happens when model-learning is hard?
  - For e.g., video, NLP (conversational AI), etc..
- Reward centric dynamics/representation learning [Du et al. 2020]
- Instance optimal bounds for offline RL
- Related problems: model learning, uncertainty quantification and planning in online RL

Conclusions

- Proposed a framework for model-based offline RL.
- Relies on supervised learning + optimization.
- Very effective
  - Learning dynamics model is feasible
  - State representation is sufficient
- Future work
  - Model selection, exploration etc.