Understanding Robust Reinforcement Learning

Han Zhong

Peking University

November 21, 2023

Robust Reinforcement Learning

What is robust reinforcement learning?

- Distributionally robust RL: training a robust policy that can perform well in perturbed environments
- Corruption robust RL: finding a good policy from the corrupted data

This talk:

- **Why** do we need distributionally robust/corruption robust RL?
- **How** to perform efficient distributionally robust/corruption robust RL?

Robust Reinforcement Learning

What is robust reinforcement learning?

- Distributionally robust RL: training a robust policy that can perform well in perturbed environments
- Corruption robust RL: finding a good policy from the corrupted data

This talk:

- **Why** do we need distributionally robust/corruption robust RL?
- **How** to perform efficient distributionally robust/corruption robust RL?

Robust Reinforcement Learning

What is robust reinforcement learning?

- Distributionally robust RL: training a robust policy that can perform well in perturbed environments
- Corruption robust RL: finding a good policy from the corrupted data

This talk:

- **Why** do we need distributionally robust/corruption robust RL?
- **How** to perform efficient distributionally robust/corruption robust RL?

Robust Reinforcement Learning

What is robust reinforcement learning?

- Distributionally robust RL: training a robust policy that can perform well in perturbed environments
- Corruption robust RL: finding a good policy from the corrupted data

This talk:

- **Why** do we need distributionally robust/corruption robust RL?
- **How** to perform efficient distributionally robust/corruption robust RL?

Robust Reinforcement Learning

What is robust reinforcement learning?

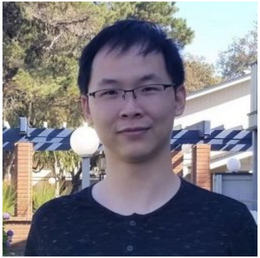
- Distributionally robust RL: training a robust policy that can perform well in perturbed environments
- Corruption robust RL: finding a good policy from the corrupted data

This talk:

- **Why** do we need distributionally robust/corruption robust RL?
- **How** to perform efficient distributionally robust/corruption robust RL?

Part 1.1: Why do we need distributionally robust RL?

Jiachen Hu
PKU



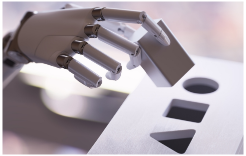
Chi Jin
Princeton

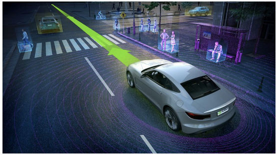
Liwei Wang
PKU

Provable Sim-to-real Transfer in Continuous Domain with Partial Observations. *International Conference on Learning Representations (ICLR) 2023.*

Offline Reinforcement Learning

Offline RL: learning optimal decisions from fixed offline datasets



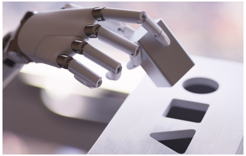


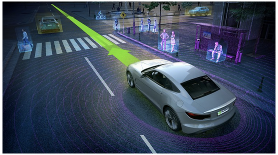
Offline RL has achieved great success in various domains, but ...

Challenge: Sim-to-Real Gap

Offline Reinforcement Learning

Offline RL: learning optimal decisions from fixed offline datasets





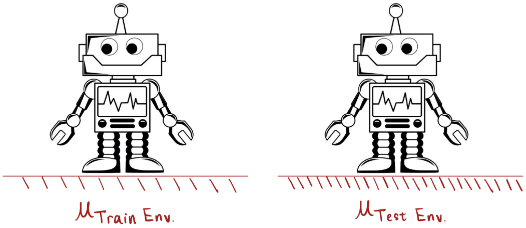
Offline RL has achieved great success in various domains, but ...

Challenge: Sim-to-Real Gap

Challenge: Sim-to-Real Gap

Example: Robotics

- Goal: Train a moving robot in a simulated environment.
- The simulated training environment has coefficient of friction $\mu_{\text{Train Env.}}$
- The environment to deploy the robot has coefficient of friction $\mu_{\text{Test Env.}} \neq \mu_{\text{Train Env.}}$
- A different moving dynamic between training and testing environments!
- Naively applying standard offline RL methods does not work.



Challenge: Sim-to-Real Gap

A general problem: mismatch between the dynamics of training and testing environments:

$$\mathbb{P}_{\text{Train Env.}}(\cdot) \neq \mathbb{P}_{\text{Test Env.}}(\cdot)$$

Non-robust offline RL methods will fail to generalize to testing environments :(

Solution: distributionally robust RL

- Takes the discrepancy between training and testing environments into account :)
- Seeks to find an optimal decision policy that is robust to the worst case testing environment.
- Mathematically, combines the framework of
  - Distributionally robust optimization (DRO)
  - Markov decision process (MDP); Linear Quadratic Regulator/Gaussian (LQR/LQG)

Challenge: Sim-to-Real Gap

A general problem: mismatch between the dynamics of training and testing environments:

$$\mathbb{P}_{\text{Train Env.}}(\cdot) \neq \mathbb{P}_{\text{Test Env.}}(\cdot)$$

Non-robust offline RL methods will fail to generalize to testing environments :(

Solution: distributionally robust RL

- Takes the discrepancy between training and testing environments into account :)
- Seeks to find an optimal decision policy that is robust to the worst case testing environment.
- Mathematically, combines the framework of
  - Distributionally robust optimization (DRO)
  - Markov decision process (MDP); Linear Quadratic Regulator/Gaussian (LQR/LQG)

Distributionally Robust RL can Efficiently Reduce the Sim-to-Real Gap

Theoretical formulation:

- Simulator class $\mathscr{E}$, i.e., a class of MDP/LQR/LQG constructed by the experimental designer.
- True environment $\Theta^* \in \mathscr{E}$.
- For a policy $\pi(\mathscr{E})$ trained from the simulator class $\mathscr{E}$, its sim-to-real gap is defined as

$$\mathrm{Gap}(\pi(\mathscr{E})) = V^*(\Theta^*) - V^{\pi(\mathscr{E})}(\Theta^*),$$

  where $V^*(\Theta^*)$ is the optimal value function and $V^{\pi(\mathscr{E})}(\Theta^*)$ is the value function of policy $\pi(\mathscr{E})$ under the environment Θ^*.
- Distributionally robust training (also known as robust adversarial training):

$$\pi_{\mathrm{robust}} = \arg\min_{\pi} \max_{\Theta \in \mathscr{E}} [V^*(\Theta) - V^{\pi}(\Theta)].$$

Distributionally Robust RL can Efficiently Reduce the Sim-to-Real Gap

Upper bound

Under certain regularity assumptions, we have

$$\mathrm{Gap}(\pi_{\mathrm{robust}}) \le \tilde{\mathcal{O}}(\sqrt{\delta_{\mathscr{E}} H}),$$

where $\delta_{\mathscr{E}}$ denotes the intrinsic complexity of simulator class $\mathscr{E}$ and H is the number of steps.

Lower Bound

Under same assumptions, for any policy π there exists a model class $\mathscr{E}$ and a choice of $\Theta^* \in \mathscr{E}$ such that:

$$\mathrm{Gap}(\pi) \ge \Omega(\sqrt{H}).$$

Distributionally robust RL reduces the sim-to-real gap efficiently (nearly optimally).

Part 1.2: How to Solve Distributionally Robust RL Efficiently?[1]

Jose Blanchet
Stanford

Miao Lu
Stanford

Tong Zhang
HKUST

Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. *Short version at Conference on Neural Information Processing Systems (NeurIPS) 2023*

[1] Most of the slides in this part are credited to Miao Lu.

A Review of Standard Offline RL

Offline RL uses the framework of **Markov decision process (MDP)**: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}^\star, R)$.

- We consider a finite-horizon decision process that ends after H decision steps.
- $\mathbb{P}^\star = \{\mathbb{P}^\star_h\}_{h\in[H]}$ and $R = \{R_h\}_{h\in[H]}$.

Interaction protocol: an agent interacts with $\mathcal{M}$ in the form of episodes (H steps). In each episode:

- at each step $h \in [H]$, the agent observes a state $s_h \in \mathcal{S}$ and takes an action $a_h \in \mathcal{A}$.
- the env. transits to $s_{h+1} \sim \mathbb{P}^\star_h(\cdot|s_h, a_h)$, and the agent receives reward $r_h = R_h(s_h, a_h)$.
- the episode ends after the agent takes the action a_H at step H.

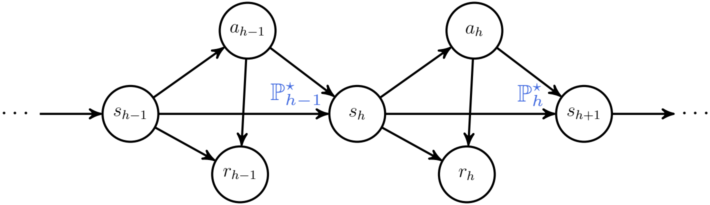
A Review of Standard Offline RL

Offline RL uses the framework of **Markov decision process (MDP)**: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}^\star, R)$.

- We consider a finite-horizon decision process that ends after H decision steps.
- $\mathbb{P}^\star = \{\mathbb{P}^\star_h\}_{h\in[H]}$ and $R = \{R_h\}_{h\in[H]}$.

Interaction protocol: an agent interacts with $\mathcal{M}$ in the form of episodes (H steps). In each episode:

- at each step $h \in [H]$, the agent observes a state $s_h \in \mathcal{S}$ and takes an action $a_h \in \mathcal{A}$.
- the env. transits to $s_{h+1} \sim \mathbb{P}^\star_h(\cdot|s_h, a_h)$, and the agent receives reward $r_h = R_h(s_h, a_h)$.
- the episode ends after the agent takes the action a_H at step H.



A Review of Standard Offline RL

Goal of offline RL: given an offline dataset $\mathcal{D}$ collected a priori, with N trajectories (episodes):

$$\mathcal{D} = \left\{ \left(s_h^\tau, a_h^\tau, r_h^\tau, s_{h+1}^\tau \right) \right\}_{h \in [H], \tau \in [N]} \quad a_h^\tau \sim \pi_h^{\mathrm{b}}(\cdot | s_h^\tau), \quad s_{h+1}^\tau \sim \mathbb{P}_h^\star(\cdot | s_h^\tau, a_h^\tau)$$

to find the optimal policy $\pi^\star = \{\pi_h\}_{h \in [H]}$ with $\pi_h : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the expected total reward:

$$\pi^\star \in \arg\max_{\pi = \{\pi_h\}_{h \in [H]} : \pi_h : \mathcal{S} \mapsto \mathcal{A}} V_1^\pi(s_1; \mathbb{P}^\star)$$

- The total reward from step h:

$$V_h^\pi(s_h; \mathbb{P}^\star) := \mathbb{E}_{\pi, \mathbb{P}^\star} \left[\sum_{h'=h}^{H} R_{h'}(s_{h'}, a_{h'}) \middle| s_h; a_{h'} \sim \pi_{h'}(\cdot | s_{h'}), s_{h'+1} \sim \mathbb{P}_{h'}^\star(\cdot | s_{h'}, a_{h'}) \right]$$

- No interaction with the real environment, only using offline data $\mathcal{D}$.
- **The policy is evaluated on the same dynamics $\mathbb{P}^\star$ as the data generation process!**

A Unified Framework of Robust Offline RL

Robust offline RL considers discrepancy between training and testing environments, and seeks to maximize the worst case expected total rewards in testing environments.

It uses the framework of **robust Markov decision process (RMDP)**, denoted by

$$\mathcal{M}_{\Phi} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}^{\star}, R, \Phi),$$

- Φ denotes the robust set (mapping) of transition dynamics,
- Interpretations of $\mathbb{P}^{\star}$ and Φ:
  - $\mathbb{P}^{\star}$: the dynamics of the training environment (the transition to generate offline data), also called the nominal transition kernel.
  - $\mathbb{P}' \in \Phi$: possible dynamics of the testing environments.
- Ususally, Φ is a "ball of distribution" centered at $\mathbb{P}^{\star}$, e.g., ϕ-divergence ball, wasserstein ball.

A Unified Framework of Robust Offline RL

Robust offline RL considers discrepancy between training and testing environments, and seeks to maximize the worst case expected total rewards in testing environments.

It uses the framework of **robust Markov decision process (RMDP)**, denoted by

$$\mathcal{M}_{\boldsymbol{\Phi}} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}^\star, R, \boldsymbol{\Phi}),$$

- $\boldsymbol{\Phi}$ denotes the robust set (mapping) of transition dynamics,
- Interpretations of $\mathbb{P}^\star$ and $\boldsymbol{\Phi}$:
  - $\mathbb{P}^\star$: the dynamics of the training environment (the transition to generate offline data), also called the nominal transition kernel.
  - $\mathbb{P}' \in \boldsymbol{\Phi}$: possible dynamics of the testing environments.
- Ususally, $\boldsymbol{\Phi}$ is a "ball of distribution" centered at $\mathbb{P}^\star$, e.g., ϕ-divergence ball, wasserstein ball.

A Unified Framework of Robust Offline RL

Goal of robust offline RL: given an offline dataset collected a prior from environment $\mathbb{P}^\star$:

$$\mathcal{D} = \Big\{ \big(s_h^\tau, a_h^\tau, r_h^\tau, s_{h+1}^\tau\big) \Big\}_{h\in[H],\tau\in[N]} \quad a_h^\tau \sim \pi_h^{\mathrm{b}}(\cdot|s_h^\tau), \quad s_{h+1}^\tau \sim \mathbb{P}_h^\star(\cdot|s_h^\tau, a_h^\tau)$$

to find the optimal robust policy $\pi^\star : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the robust expected total rewards:

$$\pi^\star \in \mathop{\arg\max}_{\pi=\{\pi_h\}_{h\in[H]}:\pi_h:\mathcal{S}\mapsto\mathcal{A}} \ \min_{\mathbb{P}'=\{\mathbb{P}'_h\}_{h\in[H]}:\mathbb{P}'_h\in\Phi(\mathbb{P}_h^\star)} V_1^\pi(s_1;\mathbb{P}')$$

- $V_1^\pi(s_h;\mathbb{P}')$ is same defined as in standard offline RL, but now $\pi^\star$ maximizes the worst case value.
- No access to data from environment $\mathbb{P}' \in \Phi$, but only the offline dataset $\mathcal{D}$ from $\mathbb{P}^\star$.

The policy is evaluated on the worst case dynamics $\mathbb{P}' \in \Phi$ of the testing environments!

A Unified Framework of Robust Offline RL

Goal of robust offline RL: given an offline dataset collected a prior from environment $\mathbb{P}^\star$:

$$\mathcal{D} = \left\{\left(s_h^\tau, a_h^\tau, r_h^\tau, s_{h+1}^\tau\right)\right\}_{h\in[H],\tau\in[N]} \quad a_h^\tau \sim \pi_h^{\mathrm{b}}(\cdot|s_h^\tau), \quad s_{h+1}^\tau \sim \mathbb{P}_h^\star(\cdot|s_h^\tau, a_h^\tau)$$

to find the optimal robust policy $\pi^\star : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the robust expected total rewards:

$$\pi^\star \in \underset{\pi=\{\pi_h\}_{h\in[H]}:\pi_h:\mathcal{S}\mapsto\mathcal{A}}{\arg\max} \; \min_{\mathbb{P}'=\{\mathbb{P}'_h\}_{h\in[H]}:\mathbb{P}'_h\in\boldsymbol{\Phi}(\mathbb{P}_h^\star)} V_1^\pi(s_1; \mathbb{P}')$$

- $V_1^\pi(s_h; \mathbb{P}')$ is same defined as in standard offline RL, but now $\pi^\star$ maximizes the worst case value.
- No access to data from environment $\mathbb{P}' \in \boldsymbol{\Phi}$, but only the offline dataset $\mathcal{D}$ from $\mathbb{P}^\star$.

The policy is evaluated on the worst case dynamics $\mathbb{P}' \in \boldsymbol{\Phi}$ of the testing environments!

A Unified Framework of Robust Offline RL

Goal of robust offline RL: given an offline dataset collected a prior from environment $\mathbb{P}^\star$:

$$\mathcal{D} = \left\{ \left(s_h^\tau, a_h^\tau, r_h^\tau, s_{h+1}^\tau \right) \right\}_{h \in [H], \tau \in [N]} \quad a_h^\tau \sim \pi_h^{\mathrm{b}}(\cdot | s_h^\tau), \quad s_{h+1}^\tau \sim \mathbb{P}_h^\star(\cdot | s_h^\tau, a_h^\tau)$$

to find the optimal robust policy $\pi^\star : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the robust expected total rewards:

$$\pi^\star \in \operatorname*{arg\,max}_{\pi = \{\pi_h\}_{h \in [H]} : \pi_h : \mathcal{S} \mapsto \mathcal{A}} \; \min_{\mathbb{P}' = \{\mathbb{P}'_h\}_{h \in [H]} : \mathbb{P}'_h \in \boldsymbol{\Phi}(\mathbb{P}_h^\star)} V_1^\pi(s_1; \mathbb{P}')$$

- $V_1^\pi(s_h; \mathbb{P}')$ is same defined as in standard offline RL, but now $\pi^\star$ maximizes the worst case value.
- No access to data from environment $\mathbb{P}' \in \boldsymbol{\Phi}$, but only the offline dataset $\mathcal{D}$ from $\mathbb{P}^\star$.

The policy is evaluated on the worst case dynamics $\mathbb{P}' \in \boldsymbol{\Phi}$ of the testing environments!

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For Q1, we identify that "**Double Pessimism**" is the desired general principle.
- For Q2, we propose the Doubly Pessimistic Model-based Policy Optimization (P^{2}MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For Q1, we identify that "**Double Pessimism**" is the desired general principle.
- For Q2, we propose the Doubly Pessimistic Model-based Policy Optimization (P^2MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For Q1, we identify that "**Double Pessimism**" is the desired general principle.
- For Q2, we propose the Doubly Pessimistic Model-based Policy Optimization (P^2MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For Q1, we identify that "**Double Pessimism**" is the desired general principle.
- For Q2, we propose the Doubly Pessimistic Model-based Policy Optimization (P^2MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For **Q1**, we identify that "**Double Pessimism**" is the desired general principle.
- For **Q2**, we propose the Doubly Pessimistic Model-based Policy Optimization (P^2MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For **Q1**, we identify that "**Double Pessimism**" is the desired general principle.
- For **Q2**, we propose the Doubly Pessimistic Model-based Policy Optimization (P^2MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

Questions:

Q1: What is the general learning principle for distributionally robust offline RL?

Q2: Based on the principle, how to design a generic algorithm for distributionally robust offline RL in the context of function approximation?

This work:

- For **Q1**, we identify that "**Double Pessimism**" is the desired general principle.
- For **Q2**, we propose the Doubly Pessimistic Model-based Policy Optimization (P^{2}MPO) algorithm framework for robust offline RL, with provable sample complexity guarantee.
- Furthermore, we extend our study to multi-agent decision making by investigating robust Markov games (RMGs).

More Detailed Setups

- Model space $\mathcal{P}_{\mathrm{M}} \subseteq \mathcal{P} := \{\mathbb{P}_h(\cdot|\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})\}$, where $\mathcal{S}$ can be infinite. It holds $\mathbb{P}_h^\star \in \mathcal{P}_{\mathrm{M}}$.
- Robust mapping $\mathbf{\Phi} : \mathcal{P}_{\mathrm{M}} \mapsto 2^{\mathcal{P}}$. E.g., $\mathbf{\Phi}(\mathbb{P}_h)$ is the robust set of $\mathbb{P}_h \in \mathcal{P}_{\mathrm{M}}$.
- Robust value functions: we define for each $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]} \subset \mathcal{P}_{\mathrm{M}}$,

$$
\begin{aligned}
V_{h,\mathbb{P},\mathbf{\Phi}}^{\pi}(s) &:= \min_{\substack{\mathbb{P}'_h \in \mathbf{\Phi}(\mathbb{P}_h) \\ 1\le h\le H}} V_h^{\pi}(s;\mathbb{P}') \\
&= \min_{\substack{\mathbb{P}'_h \in \mathbf{\Phi}(\mathbb{P}_h) \\ 1\le h\le H}} \mathbb{E}_{\pi,\mathbb{P}'}\left[\sum_{h'=h}^{H} R_{h'}(s_{h'},a_{h'}) \middle| s_h; a_{h'} \sim \pi_{h'}(\cdot|s_{h'}), s_{h'+1} \sim \mathbb{P}'_{h'}(\cdot|s_{h'},a_{h'})\right],
\end{aligned}
$$

- Formally, the goal is to find a policy $\widehat{\pi}$ from $\mathcal{D}$ that minimizes its suboptimality gap from $\pi^\star$:

$$
\mathrm{SubOpt}(\widehat{\pi}; s_1) := V_{1,\mathbb{P},\mathbf{\Phi}}^{\pi^\star}(s_1) - V_{1,\mathbb{P},\mathbf{\Phi}}^{\widehat{\pi}}(s_1),
$$

Here $\pi^\star$ is the optimal robust policy. For simplicity, we assume a fixed $s_1 \in \mathcal{S}$.

Main Challenges

Distributional shifts from two souces:

- The mismatch between the training environment dynamic $\mathbb{P}^{\star}$ and the testing environment dynamics $\mathbb{P}' \in \mathbf{\Phi}$.
  - we only have data from $\mathbb{P}^{\star}$, but we need to evaluate on distributions induced by $\mathbb{P}' \in \mathbf{\Phi}$.
- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
  - we only have data from $\pi^{\star}$, but we need to evaluate on distributions induced by learned $\widehat{\pi}$.

Large state space $\mathcal{S}$:

- The state space can be infinite in general, where existing methods for tabular RMDPs fail.

Main Challenges

Distributional shifts from two souces:

- The mismatch between the training environment dynamic $\mathbb{P}^\star$ and the testing environment dynamics $\mathbb{P}' \in \mathbf{\Phi}$.
  - we only have data from $\mathbb{P}^\star$, but we need to evaluate on distributions induced by $\mathbb{P}' \in \mathbf{\Phi}$.
- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
  - we only have data from $\pi^\star$, but we need to evaluate on distributions induced by learned $\widehat{\pi}$.

Large state space $\mathcal{S}$:

- The state space can be infinite in general, where existing methods for tabular RMDPs fail.

Main Challenges

Distributional shifts from two souces:

- The mismatch between the training environment dynamic $\mathbb{P}^\star$ and the testing environment dynamics $\mathbb{P}' \in \mathbf{\Phi}$.
  - we only have data from $\mathbb{P}^\star$, but we need to evaluate on distributions induced by $\mathbb{P}' \in \mathbf{\Phi}$.
- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
  - we only have data from $\pi^\star$, but we need to evaluate on distributions induced by learned $\widehat{\pi}$.

Large state space $\mathcal{S}$:

- The state space can be infinite in general, where existing methods for tabular RMDPs fail.

Main Challenges

Distributional shifts from two souces:

- The mismatch between the training environment dynamic $\mathbb{P}^\star$ and the testing environment dynamics $\mathbb{P}' \in \mathbf{\Phi}$.
  - we only have data from $\mathbb{P}^\star$, but we need to evaluate on distributions induced by $\mathbb{P}' \in \mathbf{\Phi}$.
- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
  - we only have data from $\pi^\star$, but we need to evaluate on distributions induced by learned $\widehat{\pi}$.

Large state space $\mathcal{S}$:

- The state space can be infinite in general, where existing methods for tabular RMDPs fail.

Pessimism: Handling Distributional Shifts

In **standard offline RL**, we have one source of distributional shift:

- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
- A naive attempt would require the data to cover the distributions induced by all possible policy $\widehat{\pi}$.
- The solution: being "pessimism" in the face of data uncertainty that originates from the statistical estimation of the transition kernel $\mathbb{P}^\star$ [Jin et al., 2020, Uehara and Sun, 2021].
- With pessimism, one can efficiently learn the optimal policy with only "partial coverage data" – only covering the trajectories induced by the optimal policy $\pi^\star$ (the mininal assumption).

Pessimism: Handling Distributional Shifts

In **standard offline RL**, we have one source of distributional shift:

- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
- A naive attempt would require the data to cover the distributions induced by all possible policy $\widehat{\pi}$.
- The solution: being "pessimism" in the face of data uncertainty that originates from the statistical estimation of the transition kernel $\mathbb{P}^\star$ [Jin et al., 2020, Uehara and Sun, 2021].
- With pessimism, one can efficiently learn the optimal policy with only "partial coverage data" – only covering the trajectories induced by the optimal policy $\pi^\star$ (the mininal assumption).

Pessimism: Handling Distributional Shifts

In **standard offline RL**, we have one source of distributional shift:

- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
- A naive attempt would require the data to cover the distributions induced by all possible policy $\widehat{\pi}$.
- The solution: being "pessimism" in the face of data uncertainty that originates from the statistical estimation of the transition kernel $\mathbb{P}^{\star}$ [Jin et al., 2020, Uehara and Sun, 2021].
- With pessimism, one can efficiently learn the optimal policy with only "partial coverage data" – only covering the trajectories induced by the optimal policy $\pi^{\star}$ (the mininal assumption).

Pessimism: Handling Distributional Shifts

In **standard offline RL**, we have one source of distributional shift:

- The mismatch between the behavior policy π^{b} and the target policies $\widehat{\pi}$ to be learned.
- A naive attempt would require the data to cover the distributions induced by all possible policy $\widehat{\pi}$.
- The solution: being "pessimism" in the face of data uncertainty that originates from the statistical estimation of the transition kernel $\mathbb{P}^{\star}$ [Jin et al., 2020, Uehara and Sun, 2021].
- With pessimism, one can efficiently learn the optimal policy with only "partial coverage data" – only covering the trajectories induced by the optimal policy $\pi^{\star}$ (the mininal assumption).

Double Pessimism: Handling Coupled Distributional Shifts

In **robust offline RL**, we have two coupled sources of distributional shift ($\mathbb{P}^\star$ vs $\mathbb{P}' \in \mathbf{\Phi}$, and $\pi^\star$ vs $\widehat{\pi}$).

- Solution: "double pessimism"
  - pessimism in the face of data uncertainty which originates from statistical estimation of the nominal transition kernel $\mathbb{P}^\star$;
  - pessimism in the face of testing env. uncertainty which comes from the target of finding a robust policy against the worst case testing env. $\mathbb{P}' \in \mathbf{\Phi}(\mathbb{P}^\star)$.
- However, $\mathbf{\Phi}(\mathbb{P})$ relies on $\mathbb{P}$.
  - perform pessimism in an iterated manner!

Double Pessimism: Handling Coupled Distributional Shifts

In **robust offline RL**, we have two coupled sources of distributional shift ($\mathbb{P}^\star$ vs $\mathbb{P}' \in \mathbf{\Phi}$, and $\pi^\star$ vs $\widehat{\pi}$).

- Solution: "double pessimism"
  - pessimism in the face of data uncertainty which originates from statistical estimation of the nominal transition kernel $\mathbb{P}^\star$;
  - pessimism in the face of testing env. uncertainty which comes from the target of finding a robust policy against the worst case testing env. $\mathbb{P}' \in \mathbf{\Phi}(\mathbb{P}^\star)$.
- However, $\mathbf{\Phi}(\mathbb{P})$ relies on $\mathbb{P}$.
  - perform pessimism in an iterated manner!

Double Pessimism: Handling Coupled Distributional Shifts

In **robust offline RL**, we have two coupled sources of distributional shift ($\mathbb{P}^\star$ vs $\mathbb{P}' \in \mathbf{\Phi}$, and $\pi^\star$ vs $\widehat{\pi}$).

- Solution: "double pessimism"
  - pessimism in the face of data uncertainty which originates from statistical estimation of the nominal transition kernel $\mathbb{P}^\star$;
  - pessimism in the face of testing env. uncertainty which comes from the target of finding a robust policy against the worst case testing env. $\mathbb{P}' \in \mathbf{\Phi}(\mathbb{P}^\star)$.
- However, $\mathbf{\Phi}(\mathbb{P})$ relies on $\mathbb{P}$.
  - perform pessimism in an iterated manner!

Double Pessimism: Handling Coupled Distributional Shifts

In **robust offline RL**, we have two coupled sources of distributional shift ($\mathbb{P}^\star$ vs $\mathbb{P}' \in \mathbf{\Phi}$, and $\pi^\star$ vs $\widehat{\pi}$).

- Solution: “double pessimism”
  - pessimism in the face of data uncertainty which originates from statistical estimation of the nominal transition kernel $\mathbb{P}^\star$;
  - pessimism in the face of testing env. uncertainty which comes from the target of finding a robust policy against the worst case testing env. $\mathbb{P}' \in \mathbf{\Phi}(\mathbb{P}^\star)$.
- However, $\mathbf{\Phi}(\mathbb{P})$ relies on $\mathbb{P}$.
  - perform pessimism in an iterated manner!

Algorithm Framework: P^2MPO

Algorithm 1: Doubly Pessimistic Model-based Policy Optimization (P^2MPO)

1. **Model estimation step:**
   Obtain a confidence region $\widehat{\mathcal{P}} = \texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ of $\mathbb{P}^\star$.

2. **Doubly pessimistic policy optimization step:**
   Set the policy $\widehat{\pi}$ as

$$\widehat{\pi} = \arg\max_{\pi} J_{\mathtt{Pess}^2}(\pi)$$

   where $J_{\mathtt{Pess}^2}(\pi)$ is defined as a doubly pessimistic value estimator:

$$J_{\mathtt{Pess}^2}(\pi) := \min_{\substack{\mathbb{P}_h \in \widehat{\mathcal{P}}_h \\ 1 \le h \le H}} \min_{\substack{\mathbb{P}'_h \in \mathbf{\Phi}(\mathbb{P}_h) \\ 1 \le h \le H}} V_1^\pi(s_1; \mathbb{P}')$$

- One can realize P^2MPO by specifying the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ for concrete RMDPs.

Algorithm Framework: P^{2}MPO

Algorithm 1: Doubly Pessimistic Model-based Policy Optimization (P^{2}MPO)

1. **Model estimation step:**
   Obtain a confidence region $\widehat{\mathcal{P}} = \texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ of $\mathbb{P}^\star$.

2. **Doubly pessimistic policy optimization step:**
   Set the policy $\widehat{\pi}$ as

$$\widehat{\pi} = \arg\max_{\pi} J_{\mathtt{Pess}^2}(\pi)$$

   where $J_{\mathtt{Pess}^2}(\pi)$ is defined as a doubly pessimistic value estimator:

$$J_{\mathtt{Pess}^2}(\pi) := \min_{\substack{\mathbb{P}_h \in \widehat{\mathcal{P}}_h \\ 1 \le h \le H}} \min_{\substack{\mathbb{P}'_h \in \Phi(\mathbb{P}_h) \\ 1 \le h \le H}} V_1^{\pi}(s_1; \mathbb{P}')$$

- One can realize P^{2}MPO by specifying the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ for concrete RMDPs.

Algorithm Framework: P^{2}MPO

Algorithm 1: Doubly Pessimistic Model-based Policy Optimization (P^{2}MPO)

1. **Model estimation step:**
   Obtain a confidence region $\widehat{\mathcal{P}} = \texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ of $\mathbb{P}^\star$.
2. **Doubly pessimistic policy optimization step:**
   Set the policy $\widehat{\pi}$ as

$$\widehat{\pi} = \arg\max_{\pi} J_{\texttt{Pess}^2}(\pi)$$

where $J_{\texttt{Pess}^2}(\pi)$ is defined as a doubly pessimistic value estimator:

$$J_{\texttt{Pess}^2}(\pi) := \min_{\substack{\mathbb{P}_h \in \widehat{\mathcal{P}}_h \\ 1 \le h \le H}} \min_{\substack{\mathbb{P}'_h \in \mathbf{\Phi}(\mathbb{P}_h) \\ 1 \le h \le H}} V_1^\pi(s_1; \mathbb{P}')$$

- One can realize P^{2}MPO by specifying the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ for concrete RMDPs.

Algorithm Framework: P^{2}MPO

Algorithm 1: Doubly Pessimistic Model-based Policy Optimization (P^{2}MPO)

1. **Model estimation step:**
   Obtain a confidence region $\widehat{\mathcal{P}} = \texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ of $\mathbb{P}^\star$.

2. **Doubly pessimistic policy optimization step:**
   Set the policy $\widehat{\pi}$ as

$$\widehat{\pi} = \arg\max_{\pi} J_{\texttt{Pess}^2}(\pi)$$

where $J_{\texttt{Pess}^2}(\pi)$ is defined as a doubly pessimistic value estimator:

$$J_{\texttt{Pess}^2}(\pi) := \min_{\substack{\mathbb{P}_h \in \widehat{\mathcal{P}}_h \\ 1 \le h \le H}} \min_{\substack{\mathbb{P}'_h \in \mathbf{\Phi}(\mathbb{P}_h) \\ 1 \le h \le H}} V_1^\pi(s_1; \mathbb{P}')$$

- One can realize P^{2}MPO by specifying the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$ for concrete RMDPs.

Two Conditions on Model Estimation Subroutine

In order to ensure sample-efficient learning of the optimal robust policy, the algorithm framework builds upon two abstract conditions on the model estimation subroutine $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$.

Condition 1 (Accuracy).

With probability at least $1-\delta$, it holds that $\mathbb{P}_h^\star \in \widehat{\mathcal{P}}_h$ for any $h \in [H]$.

- This simply means that the confidence region $\widehat{\mathcal{P}}$ needs to contain the nominal transition kernel $\mathbb{P}^\star$.

Two Conditions on Model Estimation Subroutine

Condition 2 (Robust estimation error).

For some function of the sample size N and failure probability δ denoted by $\mathrm{Err}_h^{\mathbf{\Phi}}(N,\delta) < +\infty$, with probability at least $1-\delta$, it holds that for any $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$,

$$\mathbb{E}_{(s,a)\sim d_{\mathbb{P}^\star,h}^{\pi^{\mathrm{b}}}}\left[\left(\mathcal{E}_h^{\mathbf{\Phi}}\left(s,a;\mathbb{P}_h,V_{h+1,\mathbb{P},\mathbf{\Phi}}^{\pi^\star}\right)\right)^2\right] \leq \mathrm{Err}_h^{\mathbf{\Phi}}(N,\delta)$$

where the robust estimation error is defined as

$$\mathcal{E}_h^{\mathbf{\Phi}}(s,a;\mathbb{P}_h,V) := \inf_{\mathbb{P}_h'\in\mathbf{\Phi}(\mathbb{P}_h)}\mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')] - \inf_{\mathbb{P}_h'\in\mathbf{\Phi}(\mathbb{P}_h^\star)}\mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')]$$

- This requires that each dynamic $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$ induces a small error in the sense of distributionally robust prediction between $\mathbb{P}$ and $\mathbb{P}^\star$.
- For concrete examples of RMDPs, we will implement model estimation subroutines that satisfy both Conditions 1 & 2 with $\mathrm{Err}_h^{\mathbf{\Phi}}(N,\delta) \sim \widetilde{\mathcal{O}}(1/N)$.

Two Conditions on Model Estimation Subroutine

Condition 2 (Robust estimation error).

For some function of the sample size N and failure probability δ denoted by $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta) < +\infty$, with probability at least $1 - \delta$, it holds that for any $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$,

$$\mathbb{E}_{(s,a)\sim d_{\mathbb{P}^\star,h}^{\pi^{\mathrm{b}}}}\left[\left(\mathcal{E}_h^{\mathbf{\Phi}}(s, a; \mathbb{P}_h, V_{h+1,\mathbb{P},\mathbf{\Phi}}^{\pi^\star})\right)^2\right] \leq \mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta)$$

where the robust estimation error is defined as

$$\mathcal{E}_h^{\mathbf{\Phi}}(s, a; \mathbb{P}_h, V) := \inf_{\mathbb{P}_h' \in \mathbf{\Phi}(\mathbb{P}_h)} \mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')] - \inf_{\mathbb{P}_h' \in \mathbf{\Phi}(\mathbb{P}_h^\star)} \mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')]$$

- This requires that each dynamic $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$ induces a small error in the sense of distributionally robust prediction between $\mathbb{P}$ and $\mathbb{P}^\star$.
- For concrete examples of RMDPs, we will implement model estimation subroutines that satisfy both Conditions 1 & 2 with $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta) \sim \widetilde{\mathcal{O}}(1/N)$.

Two Conditions on Model Estimation Subroutine

Condition 2 (Robust estimation error).

For some function of the sample size N and failure probability δ denoted by $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta) < +\infty$, with probability at least $1 - \delta$, it holds that for any $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$,

$$\mathbb{E}_{(s,a)\sim d_{\mathbb{P}^\star,h}^{\pi^{\mathrm{b}}}}\left[\left(\mathcal{E}_h^{\mathbf{\Phi}}(s, a; \mathbb{P}_h, V_{h+1,\mathbb{P},\mathbf{\Phi}}^{\pi^\star})\right)^2\right] \leq \mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta)$$

where the robust estimation error is defined as

$$\mathcal{E}_h^{\mathbf{\Phi}}(s, a; \mathbb{P}_h, V) := \inf_{\mathbb{P}_h' \in \mathbf{\Phi}(\mathbb{P}_h)} \mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')] - \inf_{\mathbb{P}_h' \in \mathbf{\Phi}(\mathbb{P}_h^\star)} \mathbb{E}_{s'\sim\mathbb{P}_h'(\cdot|s,a)}[V(s')]$$

- This requires that each dynamic $\mathbb{P}$ in the confidence region $\widehat{\mathcal{P}}$ induces a small error in the sense of distributionally robust prediction between $\mathbb{P}$ and $\mathbb{P}^\star$.
- For concrete examples of RMDPs, we will implement model estimation subroutines that satisfy both Conditions 1 & 2 with $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta) \sim \widetilde{\mathcal{O}}(1/N)$.

Main Assumptions

Our generic theory towards the sample efficiency of P^{2}MPO is based on two assumptions on the RMDP and the data generation process respectively.

Assumption 1 ($\mathcal{S} \times \mathcal{A}$-rectangularity).

The mapping Φ induces $\mathcal{S} \times \mathcal{A}$-rectangular robust sets: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\Phi(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}(s,a;\mathbb{P}), \quad \text{where} \quad \mathcal{P}(s,a;\mathbb{P}) \subseteq \Delta(\mathcal{S}).$$

- Interpretation: the $\mathcal{S} \times \mathcal{A}$-rectangular assumption requires the mapping $\Phi(\mathbb{P})$ gives decoupled robust sets for any $\mathbb{P}(\cdot|s,a)$ across different state-action pairs.
- We will give concrete examples of the robust set $\mathcal{P}(s,a;\mathbb{P})$ for each (s,a)-pair later.
- Discussion: P^{2}MPO can also handle other types of rectangular RMDPs (e.g., d-rectangular linear RMDP).

Main Assumptions

Our generic theory towards the sample efficiency of P^{2}MPO is based on two assumptions on the RMDP and the data generation process respectively.

Assumption 1 ($\mathcal{S} \times \mathcal{A}$-rectangularity).

The mapping $\boldsymbol{\Phi}$ induces $\mathcal{S} \times \mathcal{A}$-rectangular robust sets: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\boldsymbol{\Phi}(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}(s, a; \mathbb{P}), \quad \text{where} \quad \mathcal{P}(s, a; \mathbb{P}) \subseteq \Delta(\mathcal{S}).$$

- Interpretation: the $\mathcal{S} \times \mathcal{A}$-rectangular assumption requires the mapping $\boldsymbol{\Phi}(\mathbb{P})$ gives decoupled robust sets for any $\mathbb{P}(\cdot|s, a)$ across different state-action pairs.
- We will give concrete examples of the robust set $\mathcal{P}(s, a; \mathbb{P})$ for each (s, a)-pair later.
- Discussion: P^{2}MPO can also handle other types of rectangular RMDPs (e.g., d-rectangular linear RMDP).

Main Assumptions

Our generic theory towards the sample efficiency of P^2MPO is based on two assumptions on the RMDP and the data generation process respectively.

Assumption 1 ($\mathcal{S} \times \mathcal{A}$-rectangularity).

The mapping $\mathbf{\Phi}$ induces $\mathcal{S} \times \mathcal{A}$-rectangular robust sets: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\mathbf{\Phi}(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}(s, a; \mathbb{P}), \quad \text{where} \quad \mathcal{P}(s, a; \mathbb{P}) \subseteq \Delta(\mathcal{S}).$$

- Interpretation: the $\mathcal{S} \times \mathcal{A}$-rectangular assumption requires the mapping $\mathbf{\Phi}(\mathbb{P})$ gives decoupled robust sets for any $\mathbb{P}(\cdot|s, a)$ across different state-action pairs.
- We will give concrete examples of the robust set $\mathcal{P}(s, a; \mathbb{P})$ for each (s, a)-pair later.
- Discussion: P^2MPO can also handle other types of rectangular RMDPs (e.g., d-rectangular linear RMDP).

Main Assumptions

Our generic theory towards the sample efficiency of P^{2}MPO is based on two assumptions on the RMDP and the data generation process respectively.

Assumption 1 ($\mathcal{S} \times \mathcal{A}$-rectangularity).

The mapping $\mathbf{\Phi}$ induces $\mathcal{S} \times \mathcal{A}$-rectangular robust sets: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\mathbf{\Phi}(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}(s, a; \mathbb{P}), \quad \text{where} \quad \mathcal{P}(s, a; \mathbb{P}) \subseteq \Delta(\mathcal{S}).$$

- Interpretation: the $\mathcal{S} \times \mathcal{A}$-rectangular assumption requires the mapping $\mathbf{\Phi}(\mathbb{P})$ gives decoupled robust sets for any $\mathbb{P}(\cdot|s, a)$ across different state-action pairs.
- We will give concrete examples of the robust set $\mathcal{P}(s, a; \mathbb{P})$ for each (s, a)-pair later.
- Discussion: P^{2}MPO can also handle other types of rectangular RMDPs (e.g., d-rectangular linear RMDP).

Main Assumptions

Our generic theory towards the sample efficiency of P^{2}MPO is based on two assumptions on the RMDP and the data generation process respectively.

Assumption 1 ($\mathcal{S} \times \mathcal{A}$-rectangularity).

The mapping $\mathbf{\Phi}$ induces $\mathcal{S} \times \mathcal{A}$-rectangular robust sets: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\mathbf{\Phi}(\mathbb{P}) = \bigotimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}(s, a; \mathbb{P}), \quad \text{where} \quad \mathcal{P}(s, a; \mathbb{P}) \subseteq \Delta(\mathcal{S}).$$

- Interpretation: the $\mathcal{S} \times \mathcal{A}$-rectangular assumption requires the mapping $\mathbf{\Phi}(\mathbb{P})$ gives decoupled robust sets for any $\mathbb{P}(\cdot|s, a)$ across different state-action pairs.
- We will give concrete examples of the robust set $\mathcal{P}(s, a; \mathbb{P})$ for each (s, a)-pair later.
- Discussion: P^{2}MPO can also handle other types of rectangular RMDPs (e.g., d-rectangular linear RMDP).

Main Assumptions

$d^{\pi}_{\mathbb{P},h}(s,a)$: the state-action visitation measure at step h induced by policy π in dynamic $\mathbb{P}$.

Assumption 2 (Robust partial coverage data).

We assume that the following robust partial coverage coefficient is finite:

$$C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}} := \max_{1\leq h\leq H} \max_{\substack{\mathbb{P}_h\in\mathbf{\Phi}(\mathbb{P}^{\star}_h)\\ 1\leq h\leq H}} \mathbb{E}_{(s,a)\sim d^{\pi^{\mathrm{b}}}_{P^{\star},h}}\left[\left(\frac{d^{\pi^{\star}}_{\mathbb{P},h}(s,a)}{d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star},h}(s,a)}\right)^2\right] < +\infty, \tag{1}$$

- This only requires that the offline data $d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star}}$ can cover the trajectories induced by the optimal robust policy $d^{\pi^{\mathrm{b}}}_{\mathbb{P}}$ (for each $\mathbb{P}\in\mathbf{\Phi}(\mathbb{P}^{\star})$)!
- The robust consideration in $C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}}$ is because the policies are evaluated in a robust way in RMDPs.
- Weaker and more practice assumption than offline data from generative model or uniformly lower bounded distribution over (s,a).

Main Assumptions

$d^{\pi}_{\mathbb{P},h}(s,a)$: the state-action visitation measure at step h induced by policy π in dynamic $\mathbb{P}$.

Assumption 2 (Robust partial coverage data).

We assume that the following robust partial coverage coefficient is finite:

$$C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}} := \max_{1\leq h\leq H} \max_{\substack{\mathbb{P}_h\in\mathbf{\Phi}(\mathbb{P}^{\star}_h)\\ 1\leq h\leq H}} \mathbb{E}_{(s,a)\sim d^{\pi^{\mathrm{b}}}_{P^{\star},h}}\left[\left(\frac{d^{\pi^{\star}}_{\mathbb{P},h}(s,a)}{d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star},h}(s,a)}\right)^2\right] < +\infty, \tag{1}$$

- This only requires that the offline data $d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star}}$ can cover the trajectories induced by the optimal robust policy $d^{\pi^{\mathrm{b}}}_{\mathbb{P}}$ (for each $\mathbb{P}\in\mathbf{\Phi}(\mathbb{P}^{\star})$)!
- The robust consideration in $C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}}$ is because the policies are evaluated in a robust way in RMDPs.
- Weaker and more practice assumption than offline data from generative model or uniformly lower bounded distribution over (s,a).

Main Assumptions

$d^{\pi}_{\mathbb{P},h}(s,a)$: the state-action visitation measure at step h induced by policy π in dynamic $\mathbb{P}$.

Assumption 2 (Robust partial coverage data).

We assume that the following robust partial coverage coefficient is finite:

$$C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}} := \max_{1\leq h\leq H} \max_{\substack{\mathbb{P}_h\in\mathbf{\Phi}(\mathbb{P}^{\star}_h)\\ 1\leq h\leq H}} \mathbb{E}_{(s,a)\sim d^{\pi^{\mathrm{b}}}_{P^{\star},h}}\left[\left(\frac{d^{\pi^{\star}}_{\mathbb{P},h}(s,a)}{d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star},h}(s,a)}\right)^2\right] < +\infty, \tag{1}$$

- This only requires that the offline data $d^{\pi^{\mathrm{b}}}_{\mathbb{P}^{\star}}$ can cover the trajectories induced by the optimal robust policy $d^{\pi^{\mathrm{b}}}_{\mathbb{P}}$ (for each $\mathbb{P}\in\mathbf{\Phi}(\mathbb{P}^{\star})$)!
- The robust consideration in $C^{\star}_{\mathbb{P}^{\star},\mathbf{\Phi}}$ is because the policies are evaluated in a robust way in RMDPs.
- Weaker and more practice assumption than offline data from generative model or uniformly lower bounded distribution over (s,a).

Main Assumptions

$d^{\pi}_{\mathbb{P},h}(s,a)$: the state-action visitation measure at step h induced by policy π in dynamic $\mathbb{P}$.

Assumption 2 (Robust partial coverage data).

We assume that the following robust partial coverage coefficient is finite:

$$C^{\star}_{\mathbb{P}^\star,\mathbf{\Phi}} := \max_{1\leq h\leq H} \max_{\substack{\mathbb{P}_h\in\mathbf{\Phi}(\mathbb{P}^\star_h)\\ 1\leq h\leq H}} \mathbb{E}_{(s,a)\sim d^{\pi^{\mathrm{b}}}_{P^\star,h}}\left[\left(\frac{d^{\pi^\star}_{\mathbb{P},h}(s,a)}{d^{\pi^{\mathrm{b}}}_{\mathbb{P}^\star,h}(s,a)}\right)^2\right] < +\infty, \tag{1}$$

- This only requires that the offline data $d^{\pi^{\mathrm{b}}}_{\mathbb{P}^\star}$ can cover the trajectories induced by the optimal robust policy $d^{\pi^{\mathrm{b}}}_{\mathbb{P}}$ (for each $\mathbb{P}\in\mathbf{\Phi}(\mathbb{P}^\star)$)!
- The robust consideration in $C^{\star}_{\mathbb{P}^\star,\mathbf{\Phi}}$ is because the policies are evaluated in a robust way in RMDPs.
- Weaker and more practice assumption than offline data from generative model or uniformly lower bounded distribution over (s,a).

Main Result: Suboptimality of P^{2}MPO

Theorem 1 (Suboptimality of P^{2}MPO).

Under Assumptions 1 and 2, suppose that D^{2}MPO implements a sub-algorithm that satisfies Conditions 1 and 2, then with probability at least $1-2\delta$,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \sqrt{C^{\star}_{\mathbb{P}^\star,\boldsymbol{\Phi}}} \cdot \sum_{h=1}^{H} \sqrt{\mathrm{Err}_h^{\boldsymbol{\Phi}}(N,\delta)}.$$

- The suboptimality of P^{2}MPO is characterized by the robust partial coverage coefficient $C^{\star}_{\mathbb{P}^\star,\boldsymbol{\Phi}}$ (Assumption 2) and the sum of robust model estimation error $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N,\delta)$ (Condition 2).
- When $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N,\delta)$ achieves a rate of $\widetilde{\mathcal{O}}(N^{-1})$, P^{2}MPO enjoys a $\widetilde{\mathcal{O}}(N^{-1/2})$-suboptimality.
- In tabular setups, the robust partial coverage dependent is inevitable [Shi et al., 2022].

Main Result: Suboptimality of P^{2}MPO

Theorem 1 (Suboptimality of P^{2}MPO).

Under Assumptions 1 and 2, suppose that D^{2}MPO implements a sub-algorithm that satisfies Conditions 1 and 2, then with probability at least $1 - 2\delta$,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \sqrt{C^{\star}_{\mathbb{P}^\star, \boldsymbol{\Phi}}} \cdot \sum_{h=1}^{H} \sqrt{\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)}.$$

- The suboptimality of P^{2}MPO is characterized by the robust partial coverage coefficient $C^{\star}_{\mathbb{P}^\star, \boldsymbol{\Phi}}$ (Assumption 2) and the sum of robust model estimation error $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)$ (Condition 2).
- When $\mathrm{Err}_h^{\Phi}(N, \delta)$ achieves a rate of $\widetilde{\mathcal{O}}(N^{-1})$, P^{2}MPO enjoys a $\widetilde{\mathcal{O}}(N^{-1/2})$-suboptimality.
- In tabular setups, the robust partial coverage dependent is inevitable [Shi et al., 2022].

Main Result: Suboptimality of P^{2}MPO

Theorem 1 (Suboptimality of P^{2}MPO).

Under Assumptions 1 and 2, suppose that D^{2}MPO implements a sub-algorithm that satisfies Conditions 1 and 2, then with probability at least $1 - 2\delta$,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \sqrt{C^{\star}_{\mathbb{P}^\star, \boldsymbol{\Phi}}} \cdot \sum_{h=1}^{H} \sqrt{\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)}.$$

- The suboptimality of P^{2}MPO is characterized by the robust partial coverage coefficient $C^{\star}_{\mathbb{P}^\star, \boldsymbol{\Phi}}$ (Assumption 2) and the sum of robust model estimation error $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)$ (Condition 2).
- When $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)$ achieves a rate of $\widetilde{\mathcal{O}}(N^{-1})$, P^{2}MPO enjoys a $\widetilde{\mathcal{O}}(N^{-1/2})$-suboptimality.
- In tabular setups, the robust partial coverage dependent is inevitable [Shi et al., 2022].

Main Result: Suboptimality of P^{2}MPO

Theorem 1 (Suboptimality of P^{2}MPO).

Under Assumptions 1 and 2, suppose that D^{2}MPO implements a sub-algorithm that satisfies Conditions 1 and 2, then with probability at least $1 - 2\delta$,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \sqrt{C^{\star}_{\mathbb{P}^{\star},\boldsymbol{\Phi}}} \cdot \sum_{h=1}^{H} \sqrt{\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)}.$$

- The suboptimality of P^{2}MPO is characterized by the robust partial coverage coefficient $C^{\star}_{\mathbb{P}^{\star},\boldsymbol{\Phi}}$ (Assumption 2) and the sum of robust model estimation error $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)$ (Condition 2).
- When $\mathrm{Err}_h^{\boldsymbol{\Phi}}(N, \delta)$ achieves a rate of $\widetilde{\mathcal{O}}(N^{-1})$, P^{2}MPO enjoys a $\widetilde{\mathcal{O}}(N^{-1/2})$-suboptimality.
- In tabular setups, the robust partial coverage dependent is inevitable [Shi et al., 2022].

Our theory applies to most of known tractable RMDPs for robust offline RL and **new** models by:

- implementing the model estimation subroutine $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$;
- specifying the robust model estimation error $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta)$.

	Zhou et al. [2021]	Shi and Chi [2022]	Ma et al. [2022]	This Work
$\mathcal{S} \times \mathcal{A}$-rectangular tabular RMDP	✓!	✓	✗	✓
d-rectangular linear RMDP	✗	✗	✓	✓
$\mathcal{S} \times \mathcal{A}$-rectangular factored RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular neural RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular general RMG	NA	NA	NA	✓

Table: ✓: can tackle this model with robust partial coverage data, ✓!: requires full coverage data to solve the model, ✗: cannot tackle the model.

The yellow line denotes the models that are first proposed or proved tractable in this work.

Our theory applies to most of known tractable RMDPs for robust offline RL and **new** models by:

- implementing the model estimation subroutine $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$;
- specifying the robust model estimation error $\mathrm{Err}_h^{\mathbf{\Phi}}(N, \delta)$.

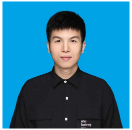
	Zhou et al. [2021]	Shi and Chi [2022]	Ma et al. [2022]	This Work
$\mathcal{S} \times \mathcal{A}$-rectangular tabular RMDP	✓!	✓	✗	✓
d-rectangular linear RMDP	✗	✗	✓	✓
$\mathcal{S} \times \mathcal{A}$-rectangular factored RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular neural RMDP	✗	✗	✗	✓
$\mathcal{S} \times \mathcal{A}$-rectangular general RMG	NA	NA	NA	✓

Table: ✓: can tackle this model with robust partial coverage data, ✓!: requires full coverage data to solve the model, ✗: cannot tackle the model.

The yellow line denotes the models that are first proposed or proved tractable in this work.

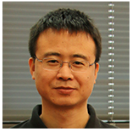
Part 2: Corruption Robust Reinforcement Learning

Rui Yang
HKUST



Jiawei Xu
CUHKSZ

Amy Zhang
UT Austin



Chongjie Zhang
WUSTL

Lei Han
Tencent

Tong Zhang
HKUST

Towards Robust Offline Reinforcement Learning under Diverse Data Corruption. *Preprint*

Offline RL with Corruption Data

Value functions in discounted MDPs:

$$V^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0 = s\right], \quad Q^\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^\pi(s')\right]$$

Goal: Find an optimal policy $\pi^* = \arg\max_\pi \mathbb{E}_{s_0\sim\rho_0}[V^\pi(s_0)]$, where ρ_0 is the initial distribution.

- Clean data (s,a,r,s'): $(s,a)\sim\mu(\cdot,\cdot)$, $r = r(s,a)$, and $s' \sim P(\cdot \mid s,a)$, where $\mu(\cdot,\cdot)$ is the fixed behavior policy. Let $\pi_\mu(a \mid s)$ denote the conditional distribution.
- Corrupted data $\mathcal{D} = \{(s_i,a_i,r_i,s_i')\}_{i=1}^N$: $(s_i,a_i)\sim\tilde{\mu}(\cdot,\cdot)$, $r = \tilde{r}(s_i,a_i)$, and $s_i' \sim \tilde{P}(\cdot \mid s_i,a_i)$. Let $\pi_{\mathcal{D}}(a \mid s)$ denote the conditional distribution.

Definition (Cumulative Corruption)

Let $\zeta = \sum_{i=1}^N(2\zeta_i + \log\zeta_i')$ denote the cumulative corruption level, where ζ_i and ζ_i' are defined as

$$\|[\mathcal{T}V](s_i,a) - [\tilde{\mathcal{T}}V](s_i,a)\|_\infty \le \zeta_i, \quad \max\left\{\frac{\pi_{\mathcal{D}}(a \mid s_i)}{\pi_\mu(a \mid s_i)}, \frac{\pi_\mu(a \mid s_i)}{\pi_{\mathcal{D}}(a \mid s_i)}\right\} \le \zeta_i', \quad \forall a \in A.$$

Offline RL with Corruption Data

Value functions in discounted MDPs:

$$V^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0 = s\right], \quad Q^\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^\pi(s')\right]$$

Goal: Find an optimal policy $\pi^* = \arg\max_\pi \mathbb{E}_{s_0\sim\rho_0}[V^\pi(s_0)]$, where ρ_0 is the initial distribution.

- Clean data (s,a,r,s'): $(s,a)\sim\mu(\cdot,\cdot)$, $r = r(s,a)$, and $s' \sim P(\cdot \mid s,a)$, where $\mu(\cdot,\cdot)$ is the fixed behavior policy. Let $\pi_\mu(a \mid s)$ denote the conditional distribution.
- Corrupted data $\mathcal{D} = \{(s_i,a_i,r_i,s_i')\}_{i=1}^N$: $(s_i,a_i)\sim\tilde{\mu}(\cdot,\cdot)$, $r = \tilde{r}(s_i,a_i)$, and $s_i' \sim \tilde{P}(\cdot \mid s_i,a_i)$. Let $\pi_{\mathcal{D}}(a \mid s)$ denote the conditional distribution.

Definition (Cumulative Corruption)

Let $\zeta = \sum_{i=1}^N(2\zeta_i + \log\zeta_i')$ denote the cumulative corruption level, where ζ_i and ζ_i' are defined as

$$\|[\mathcal{T}V](s_i,a) - [\tilde{\mathcal{T}}V](s_i,a)\|_\infty \le \zeta_i, \quad \max\left\{\frac{\pi_{\mathcal{D}}(a \mid s_i)}{\pi_\mu(a \mid s_i)}, \frac{\pi_\mu(a \mid s_i)}{\pi_{\mathcal{D}}(a \mid s_i)}\right\} \le \zeta_i', \quad \forall a \in A.$$

Implicit Q-Learning

IQL [Kostrikov et al., 2021] employs expectile regression to learn the value function:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a)\right)^2\right],$$
$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\mathcal{L}_2^\tau(Q_\theta(s,a) - V_\psi(s))\right], \quad \mathcal{L}_2^\tau(x) = |\tau - \mathbf{1}(x<0)|x^2.$$

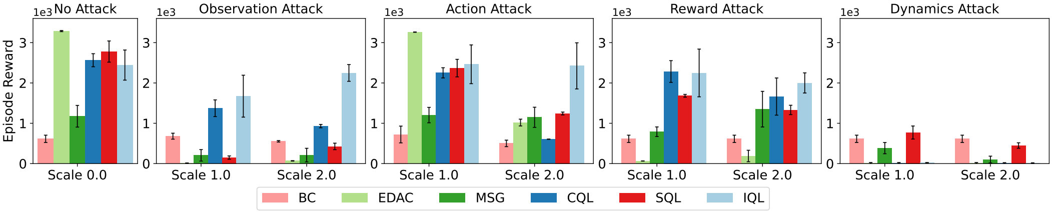
IQL further extracts the policy using weighted imitation learning with a hyperparameter β:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\exp(\beta\cdot A(s,a))\log\pi_\phi(a|s)], \quad A(s,a) = Q_\theta(s,a) - V_\psi(s).$$

Key observation:

IQL adopts the supervised policy learning instead of value-based policy learning.

Performance of IQL under Diverse Data Corruption



- Performance of offline RL algorithms under random attacks on the Hopper task.
- IQL demonstrates superior resilience to 3 out of 4 types of data corruption.

Key observation:

Supervised policy learning is more robust than value-based policy learning!

Theoretical Guarantee

Let π_{IQL} and $\tilde{\pi}_{\mathrm{IQL}}$ be the learned policy by IQL with clean data and corrupted data, respectively.

Theorem

Assuming certain partial-coverage-type assumption is satisfied with coefficient M, it holds that

$$V^{\pi_{\mathrm{IQL}}} - V^{\tilde{\pi}_{\mathrm{IQL}}} \leq \frac{\sqrt{2M}R_{\max}}{(1-\gamma)^2}\left[\sqrt{\epsilon_1} + \sqrt{\epsilon_2}\right] + \frac{2R_{\max}}{(1-\gamma)^2}\sqrt{\frac{M\zeta}{N}},$$

where ϵ_1 and ϵ_2 are imitation errors, ζ is the cumulative corruption, and N is the dataset size.

- Here ϵ_1 and ϵ_2 are imitation errors, which typically decay to zero as N goes to infinity.
- The corruption error term (second term) diminishes when $\zeta = o(N)$.
- Compared with LSVI-type algorithms:
  - Provably efficient under diverse data corruption;
  - Only requires $\zeta = o(N)$ instead of $\zeta = o(\sqrt{N})$;

Theoretical Guarantee

Let π_{IQL} and $\tilde{\pi}_{\mathrm{IQL}}$ be the learned policy by IQL with clean data and corrupted data, respectively.

Theorem

Assuming certain partial-coverage-type assumption is satisfied with coefficient M, it holds that

$$V^{\pi_{\mathrm{IQL}}} - V^{\tilde{\pi}_{\mathrm{IQL}}} \leq \frac{\sqrt{2M} R_{\max}}{(1-\gamma)^2}[\sqrt{\epsilon_1} + \sqrt{\epsilon_2}] + \frac{2R_{\max}}{(1-\gamma)^2}\sqrt{\frac{M\zeta}{N}},$$

where ϵ_1 and ϵ_2 are imitation errors, ζ is the cumulative corruption, and N is the dataset size.

- Here ϵ_1 and ϵ_2 are imitation errors, which typically decay to zero as N goes to infinity.
- The corruption error term (second term) diminishes when $\zeta = o(N)$.
- Compared with LSVI-type algorithms:
  - Provably efficient under diverse data corruption;
  - Only requires $\zeta = o(N)$ instead of $\zeta = o(\sqrt{N})$;

Theoretical Guarantee

Let π_{IQL} and $\tilde{\pi}_{\mathrm{IQL}}$ be the learned policy by IQL with clean data and corrupted data, respectively.

Theorem

Assuming certain partial-coverage-type assumption is satisfied with coefficient M, it holds that

$$V^{\pi_{\mathrm{IQL}}} - V^{\tilde{\pi}_{\mathrm{IQL}}} \leq \frac{\sqrt{2M}R_{\max}}{(1-\gamma)^2}[\sqrt{\epsilon_1} + \sqrt{\epsilon_2}] + \frac{2R_{\max}}{(1-\gamma)^2}\sqrt{\frac{M\zeta}{N}},$$

where ϵ_1 and ϵ_2 are imitation errors, ζ is the cumulative corruption, and N is the dataset size.

- Here ϵ_1 and ϵ_2 are imitation errors, which typically decay to zero as N goes to infinity.
- The corruption error term (second term) diminishes when $\zeta = o(N)$.
- Compared with LSVI-type algorithms:
  - Provably efficient under diverse data corruption;
  - Only requires $\zeta = o(N)$ instead of $\zeta = o(\sqrt{N})$;

Theoretical Guarantee

Let π_{IQL} and $\tilde{\pi}_{\mathrm{IQL}}$ be the learned policy by IQL with clean data and corrupted data, respectively.

Theorem

Assuming certain partial-coverage-type assumption is satisfied with coefficient M, it holds that

$$V^{\pi_{\mathrm{IQL}}} - V^{\tilde{\pi}_{\mathrm{IQL}}} \leq \frac{\sqrt{2M} R_{\max}}{(1-\gamma)^2}[\sqrt{\epsilon_1} + \sqrt{\epsilon_2}] + \frac{2R_{\max}}{(1-\gamma)^2}\sqrt{\frac{M\zeta}{N}},$$

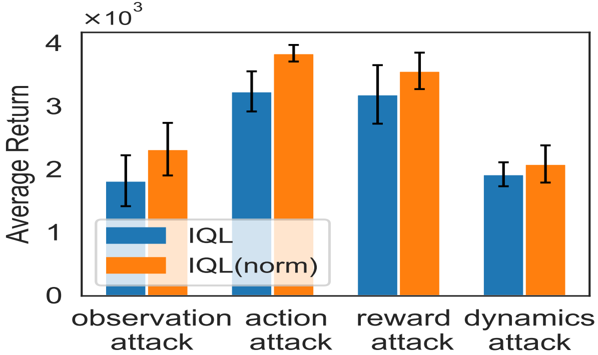
where ϵ_1 and ϵ_2 are imitation errors, ζ is the cumulative corruption, and N is the dataset size.

- Here ϵ_1 and ϵ_2 are imitation errors, which typically decay to zero as N goes to infinity.
- The corruption error term (second term) diminishes when $\zeta = o(N)$.
- Compared with LSVI-type algorithms:
  - Provably efficient under diverse data corruption;
  - Only requires $\zeta = o(N)$ instead of $\zeta = o(\sqrt{N})$;

Improvement 1: Observation Normalization

$$s_i = \frac{(s_i - \mu_o)}{\sigma_o}, \quad s_i' = \frac{(s_i' - \mu_o)}{\sigma_o},$$

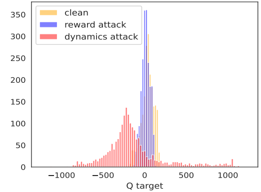
$$\mu_o = \frac{1}{2N}\sum_{i=1}^{N}(s_i + s_i'), \quad \sigma_o^2 = \frac{1}{2N}\sum_{i=1}^{N}[(s_i - \mu_o)^2 + (s_i' - \mu_o)^2].$$



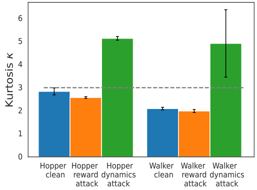
Improvement 2: Huber Loss

- Identify the heavy-tailed issue in the dynamics attack.
- Use the Huber regression

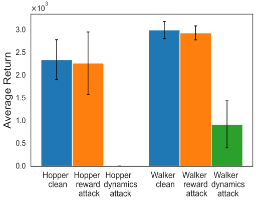
$$\mathcal{L}_Q = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[l_H^\delta(r+\gamma V(s') - Q(s,a))\right], \quad \text{where } l_H^\delta(x) = \begin{cases} \frac{1}{2\delta}x^2, & \text{if } |x| \le \delta \\ |x| - \frac{1}{2}\delta, & \text{if } |x| > \delta \end{cases}.$$



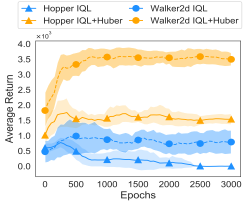
(m)



(n)



(o)



(p)

Improvement 3: Penalizing Corrupted Data via In-dataset Uncertainty

- Train K independent Q-functions $\{Q_{\theta_i}\}_{i=1}^{K}$. Let Q_α be the α-quantile value.

$$\mathcal{L}_Q(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[l_H^\delta(r + \gamma V_\psi(s') - Q_{\theta_i}(s,a))],$$

- Learn V-function based on Q_α

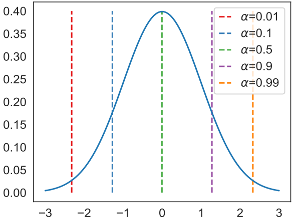
$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\mathcal{L}_2^\tau(Q_\alpha(s,a) - V_\psi(s))\right], \quad \mathcal{L}_2^\tau(x) = |\tau - \mathbf{1}(x<0)|x^2.$$

- The policy is learned to maximize the α-quantile advantage-weighted imitation learning objective:

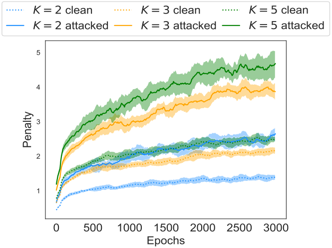
$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp(\beta A_\alpha(s,a))\log\pi_\phi(a|s)\right], \quad A_\alpha(s,a) = Q_\alpha(s,a) - V_\psi(s).$$

Improvement 3: Penalizing Corrupted Data via In-dataset Uncertainty

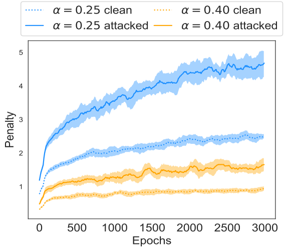
Key Insight: penalizing corrupted data via in-dataset uncertainty.



(q)



(r)



(s)

Figure: (q) Quantiles of a normal distribution. In-dataset penalty for attacked data and clean data across (r) different ensemble sizes K and (s) different quantile values α.

Robust IQL

Algorithm Robust IQL algorithm

1: Initialize policy π_ϕ and value function V_ψ, $\{Q_{\theta_i}\}_{i=1}^K$
2: Normalize the observation;
3: **for** training step$= 1, 2, \ldots, T$ **do**
4: Update value function V_ψ to minimize

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\mathcal{L}_2^\tau(Q_\alpha(s,a) - V_\psi(s))\right];$$

5: Update $\{Q_{\theta_i}\}_{i=1}^K$ independently to minimize

$$\mathcal{L}_Q(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[l_H^\delta(r + \gamma V_\psi(s') - Q_{\theta_i}(s,a))];$$

6: Update policy π_ϕ to maximize

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp(\beta A_\alpha(s,a)) \log \pi_\phi(a|s)\right]$$

Performance Under Random Corruption

Environment	Attack Element	BC	EDAC	MSG	CQL	SQL	IQL	RIQL (ours)
Halfcheetah	observation	**33.4±1.8**	2.1±0.5	-0.2±2.2	9.0±7.5	4.1±1.4	21.4±1.9	27.3±2.4
	action	36.2±0.3	47.4±1.3	**52.0±0.9**	19.9±21.3	42.9±0.4	42.2±1.9	42.9±0.6
	reward	35.8±0.9	38.6±0.3	17.5±16.4	32.6±19.6	41.7±0.8	42.3±0.4	**43.6±0.6**
	dynamics	35.8±0.9	1.5±0.2	1.7±0.4	29.2±4.0	35.5±0.4	36.7±1.8	**43.1±0.2**
Walker2d	observation	9.6±3.9	-0.2±0.3	-0.4±0.1	19.4±1.6	0.6±1.0	27.2±5.1	**28.4±7.7**
	action	18.1±2.1	83.2±1.9	25.3±10.6	62.7±7.2	76.0±4.2	71.3±7.8	**84.6±3.3**
	reward	16.0±7.4	4.3±3.6	18.4±9.5	69.4±7.4	33.8±13.8	65.3±8.4	**83.2±2.6**
	dynamics	16.0±7.4	-0.1±0.0	7.4±3.7	-0.2±0.1	15.3±2.2	17.7±7.3	**78.2±1.8**
Hopper	observation	21.5±2.9	1.0±0.5	6.9±5.0	42.8±7.0	5.2±1.9	52.0±16.6	**62.4±1.8**
	action	22.8±7.0	**100.8±0.5**	37.6±6.5	69.8±4.5	73.4±7.3	76.3±15.4	90.6±5.6
	reward	19.5±3.4	2.6±0.7	24.9±4.3	70.8±8.9	52.3±1.7	69.7±18.8	**84.8±13.1**
	dynamics	19.5±3.4	0.8±0.0	12.4±4.9	0.8±0.0	24.3±5.6	1.3±0.5	**51.5±8.1**
Average score ↑		23.7	23.5	17.0	35.5	33.8	43.6	**60.0**
Average degradation percentage ↓		0.4%	68.5%	61.5%	42.3%	45.0%	31.2%	17.0%

Performance Under Adversarial Corruption

Environment	Attack Element	BC	EDAC	MSG	CQL	SQL	IQL	RIQL (ours)
Halfcheetah	observation	34.5±1.5	1.1±0.3	1.1±0.2	5.0±11.6	8.3±0.9	32.6±2.7	**35.7±4.2**
	action	14.0±1.1	32.7±0.7	**37.3±0.7**	-2.3±1.2	32.7±1.0	27.5±0.3	31.7±1.7
	reward	35.8±0.9	40.3±0.5	**47.7±0.4**	-1.7±0.3	42.9±0.1	42.6±0.4	44.1±0.8
	dynamics	35.8±0.9	-1.3±0.1	-1.5±0.0	-1.6±0.0	10.4±2.6	26.7±0.7	**35.8±2.1**
Walker2d	observation	12.7±5.9	-0.0±0.1	2.9±2.7	61.8±7.4	1.8±1.9	37.7±13.0	**70.0±5.3**
	action	5.4±0.4	41.9±24.0	5.4±0.9	27.0±7.5	31.3±8.8	27.5±0.6	**66.1±4.6**
	reward	16.0±7.4	57.3±33.2	9.6±4.9	67.0±6.1	78.1±2.0	73.5±4.85	**85.0±1.5**
	dynamics	16.0±7.4	4.3±0.9	0.1±0.2	3.9±1.4	2.7±1.9	-0.1±0.1	**60.6±21.8**
Hopper	observation	21.6±7.1	36.2±16.2	16.0±2.8	**78.0±6.5**	8.2±4.7	32.8±6.4	50.8±7.6
	action	15.5±2.2	25.7±3.8	23.0±2.1	32.2±7.6	30.0±0.4	37.9±4.8	**63.6±7.3**
	reward	19.5±3.4	21.2±1.9	22.6±2.8	49.6±12.3	57.9±4.8	57.3±9.7	**65.8±9.8**
	dynamics	19.5±3.4	0.6±0.0	0.6±0.0	0.6±0.0	18.9±12.6	1.3±1.1	**65.7±21.1**
Average score ↑		20.5	21.7	13.7	26.6	25.8	33.1	**56.2**
Average degradation percentage ↓		13.4%	71.2%	69.9%	66.8%	57.5%	46.0%	22.0%

Conclusion

- Distributionally robust RL
  - Training a robust policy that can perform well in perturbed environments.
  - Distributionally robust RL can efficiently reduce the sim-to-real gap.
  - General learning principle for distributionally robust offline RL — double pessimism.
- Corruption robust RL
  - Finding a good policy from the corrupted data.
  - Supervised policy learning (IQL) is more robust than value-based policy optimization.
  - Robust IQL: observation normalization, Huber regression, and penalizing corrupted data via in-dataset uncertainty.

Thank you!

https://hanzhong-ml.github.io/

Conclusion

- Distributionally robust RL
  - Training a robust policy that can perform well in perturbed environments.
  - Distributionally robust RL can efficiently reduce the sim-to-real gap.
  - General learning principle for distributionally robust offline RL — double pessimism.
- Corruption robust RL
  - Finding a good policy from the corrupted data.
  - Supervised policy learning (IQL) is more robust than value-based policy optimization.
  - Robust IQL: observation normalization, Huber regression, and penalizing corrupted data via in-dataset uncertainty.

Thank you!

https://hanzhong-ml.github.io/

Conclusion

- Distributionally robust RL
  - Training a robust policy that can perform well in perturbed environments.
  - Distributionally robust RL can efficiently reduce the sim-to-real gap.
  - General learning principle for distributionally robust offline RL — double pessimism.
- Corruption robust RL
  - Finding a good policy from the corrupted data.
  - Supervised policy learning (IQL) is more robust than value-based policy optimization.
  - Robust IQL: observation normalization, Huber regression, and penalizing corrupted data via in-dataset uncertainty.

Thank you!

https://hanzhong-ml.github.io/

References I

Q. Cai, Z. Yang, C. Szepesvari, and Z. Wang. Optimistic policy optimization with general function approximations. 2020.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In Conference on Learning Theory, pages 2137–2143. PMLR, 2020.

M. Kearns and D. Koller. Efficient reinforcement learning in factored mdps. In IJCAI, volume 16, pages 740–747, 1999.

I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. arXiv preprint arXiv:2110.06169, 2021.

C. J. Li, D. Zhou, Q. Gu, and M. I. Jordan. Learning two-player mixture markov games: Kernel function approximation and correlated equilibrium. arXiv preprint arXiv:2208.05363, 2022.

X. Ma, Z. Liang, L. Xia, J. Zhang, J. Blanchet, M. Liu, Q. Zhao, and Z. Zhou. Distributionally robust offline reinforcement learning with linear function approximation. arXiv preprint arXiv:2209.06620, 2022.

References II

L. Shi and Y. Chi. Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. arXiv preprint arXiv:2208.05767, 2022.

L. Shi, G. Li, Y. Wei, Y. Chen, and Y. Chi. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. arXiv preprint arXiv:2202.13890, 2022.

M. Uehara and W. Sun. Pessimistic model-based offline reinforcement learning under partial coverage. arXiv preprint arXiv:2107.06226, 2021.

Z. Yang, C. Jin, Z. Wang, M. Wang, and M. Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. Advances in Neural Information Processing Systems, 33:13903–13916, 2020.

Z. Zhou, Z. Zhou, Q. Bai, L. Qiu, J. Blanchet, and P. Glynn. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In International Conference on Artificial Intelligence and Statistics, pages 3331–3339. PMLR, 2021.

Backup Slides

Example I: $\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP

Consider an RMDP with transition kernel parametrized by a reproducing kernel Hilbert space (RKHS).

- Model space $\mathcal{P}_{\mathrm{M}}$: let $\mathcal{H}$ be an RKHS associated with a positive definite kernel $\mathcal{K} : (\mathcal{S} \times \mathcal{A} \times \mathcal{S}) \times (\mathcal{S} \times \mathcal{A} \times \mathcal{S}) \mapsto \mathbb{R}_+$, whose feature mapping is $\boldsymbol{\psi} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathcal{H}$, then

$$\mathcal{P}_{\mathrm{M}} = \left\{ \mathbb{P}(s'|s,a) = \langle \boldsymbol{\psi}(s,a,s'), \boldsymbol{f} \rangle_{\mathcal{H}} : \boldsymbol{f} \in \mathcal{H}, \|\boldsymbol{f}\|_{\mathcal{H}} \leq B_{\mathrm{K}} \right\}.$$

- Robust mapping $\boldsymbol{\Phi}$: for any $\mathbb{P} \in \mathcal{P}_{\mathrm{M}}$,

$$\boldsymbol{\Phi}(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_{\rho}(s,a;\mathbb{P}), \quad \text{with} \quad \mathcal{P}_{\rho}(s,a;\mathbb{P}) = \left\{ \widetilde{\mathbb{P}}(\cdot) \in \Delta(\mathcal{S}) : D(\widetilde{\mathbb{P}}(\cdot) \| \mathbb{P}(\cdot|s,a)) \leq \rho \right\},$$

  - In this work, we consider $D(\cdot\|\cdot)$ as TV-distance or KL-divergence.
  - Robust counterpart of kernel MDP [Yang et al., 2020, Cai et al., 2020, Li et al., 2022].
  - Covers $\mathcal{S} \times \mathcal{A}$-rectangular tabular/linear MDPs as special cases.

Example I: $\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP

To apply algorithm D^{2}MPO and the theory, we need to specify (i) the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$, (ii) the model estimation error function $\mathrm{Err}_h^{\mathbf{\Phi}}(n, \delta)$.

Subalgorithm: model estimation I

Using the offline data $\mathcal{D}$, we first construct the maximum likelihood estimator of $\mathbb{P}^\star$:

$$\widehat{\mathbb{P}}_h = \underset{\mathbb{P} \in \mathcal{P}_{\mathrm{M}}}{\arg\max} \frac{1}{N} \sum_{\tau=1}^{N} \log \mathbb{P}(s_{h+1}^\tau | s_h^\tau, a_h^\tau).$$

After, we construct a confidence region $\widehat{\mathcal{P}}$ for the MLE estimator,

$$\widehat{\mathcal{P}}_h = \left\{ \mathbb{P} \in \mathcal{P}_{\mathrm{M}} : \frac{1}{N} \sum_{\tau=1}^{N} \|\widehat{\mathbb{P}}_h(\cdot | s_h^\tau, a_h^\tau) - \mathbb{P}(\cdot | s_h^\tau, a_h^\tau)\|_1^2 \leq \xi \right\},$$

where $\xi > 0$ is a tuning parameter controlling the size of $\widehat{\mathcal{P}}_h$. We let $\texttt{ModelEst}(\mathbb{D}, \mathcal{P}_{\mathrm{M}}) = \{\widehat{\mathcal{P}}_h\}_{h=1}^H$.

Example I: $\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP

Assumption 3: regularity of RKHS (informal)

The kernel $\mathcal{K}$ of the RKHS satisfies boundedness and exponential eigenvalue decay ($\lambda_j \lesssim \exp(-j^\gamma)$).

Corollary: suboptimality of D^{2}MPO for $\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP

Under Assumptions 1, 2, 3, by proper choosing the tuning parameter ξ, the suboptimality of D^{2}MPO for $\mathcal{S} \times \mathcal{A}$-rectangular kernel RMDP is

- when D is the TV-distance,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \mathcal{O}\left(H^2 \log(1/\gamma) \cdot \sqrt{C^\star_{\mathbb{P}^\star, \boldsymbol{\Phi}}/\gamma \cdot \log^{1+1/\gamma}(NH\mathtt{Vol}(\mathcal{S})/\delta)/N}\right),$$

- when D is the KL-divergence,

$$\mathrm{SubOpt}(\widehat{\pi}; s_1) \leq \mathcal{O}\left(H^2 \exp(H) \log(1/\gamma)/\rho \cdot \sqrt{C^\star_{\mathbb{P}^\star, \boldsymbol{\Phi}}/\gamma \cdot \log^{1+1/\gamma}(NH\mathtt{Vol}(\mathcal{S})/\delta)/N}\right).$$

Example II: $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP

Consider a tabular RMDP with factored transition kernel $\mathbb{P}_h^\star(s'|s,a) = \prod_{i=1}^d \mathbb{P}_{h,i}^\star(s'[i]|s[\mathrm{pa}_i], a)$.

- Model space $\mathcal{P}_{\mathrm{M}}$: let $\mathcal{S} = \mathcal{O}^d$, $s = (s[1], \cdots, s[d])$ and $s[i]$ is determined by $(s[\mathrm{pa}_i], a)$

$$\mathcal{P}_{\mathrm{M}} = \left\{ \mathbb{P}(s'|s,a) = \prod_{i=1}^d \mathbb{P}_i(s'[i]|s[\mathrm{pa}_i], a) \ : \ \mathbb{P}_i : \mathcal{S}[\mathrm{pa}_i] \times \mathcal{A} \mapsto \Delta(\mathcal{O}), \forall i \in [d] \right\}.$$

- Robust mapping $\mathbf{\Phi}$: for any $\mathbb{P}(s'|s,a) = \prod_{i=1}^d \mathbb{P}_{h,i}(s'[i]|s[\mathrm{pa}_i], a) \in \mathcal{P}_{\mathrm{M}}$,

$$\mathbf{\Phi}(\mathbb{P}) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_{\mathrm{Fac},\rho}(s,a;P), \quad \text{with}$$

$$\mathcal{P}_{\mathrm{Fac},\rho}(s,a;\mathbb{P}) = \left\{ \prod_{i=1}^d \widetilde{\mathbb{P}}_i(\cdot) : \widetilde{\mathbb{P}}_i(\cdot) \in \Delta(\mathcal{O}), D(\widetilde{\mathbb{P}}_i(\cdot) \| \mathbb{P}_i(\cdot|s[\mathrm{pa}_i], a)) \le \rho_i, \forall i \in [d] \right\}.$$

  - Also, we consider $D(\cdot\|\cdot)$ as TV-distance or KL-divergence.
  - Robust counterpart of factored MDP [Kearns and Koller, 1999].
  - How to utilize the factored structure to improve sample complexity?

Example II: $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP

To apply algorithm D^{2}MPO and the theory, we need to specify (i) the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$, (ii) the model estimation error function $\mathrm{Err}_h^{\mathbf{\Phi}}(n, \delta)$.

Subalgorithm: model estimation II

Using the offline data $\mathcal{D}$, we first construct the maximum likelihood estimator of each factor $\mathbb{P}_i^\star$:

$$\widehat{\mathbb{P}}_{h,i} = \mathop{\arg\max}_{\mathbb{P}_i : \mathcal{S}[\mathrm{pa}_i] \times \mathcal{A} \mapsto \Delta(\mathcal{O})} \frac{1}{N} \sum_{k=1}^{N} \log \mathbb{P}_i(s_{h+1}^\tau[i] | s_h^\tau[\mathrm{pa}_i], a_h^\tau).$$

After, we construct a confidence region $\widehat{\mathcal{P}}$ based on the MLE of each factor as

$$\hat{\mathcal{P}}_h = \left\{ P(s'|s,a) = \prod_{i=1}^{d} P_i(s'[i]|s[\mathrm{pa}_i], a) : \frac{1}{n} \sum_{i=1}^{N} \|(P_i - \hat{P}_{h,i})(\cdot|s_h^\tau[\mathrm{pa}_i], a_h^\tau)\|_1^2 \leq \xi_i, \forall i \in [d] \right\}.$$

where $\xi_i > 0$ are tuning parameters controlling the size of $\widehat{\mathcal{P}}_h$. We let $\texttt{ModelEst}(\mathbb{D}, \mathcal{P}_{\mathrm{M}}) = \{\widehat{\mathcal{P}}_h\}_{h=1}^{H}$.

Example II: $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP

To apply algorithm D^{2}MPO and the theory, we need to specify (i) the subalgorithm $\texttt{ModelEst}(\mathcal{D}, \mathcal{P}_{\mathrm{M}})$, (ii) the model estimation error function $\mathrm{Err}_h^{\boldsymbol{\Phi}}(n, \delta)$.

Subalgorithm: model estimation II

Using the offline data $\mathcal{D}$, we first construct the maximum likelihood estimator of each factor $\mathbb{P}_i^\star$:

$$\widehat{\mathbb{P}}_{h,i} = \mathop{\arg\max}_{\mathbb{P}_i : \mathcal{S}[\mathrm{pa}_i] \times \mathcal{A} \mapsto \Delta(\mathcal{O})} \frac{1}{N} \sum_{k=1}^{N} \log \mathbb{P}_i(s_{h+1}^\tau[i] | s_h^\tau[\mathrm{pa}_i], a_h^\tau).$$

After, we construct a confidence region $\widehat{\mathcal{P}}$ based on the MLE of each factor as

$$\hat{\mathcal{P}}_h = \left\{ P(s'|s,a) = \prod_{i=1}^{d} P_i(s'[i]|s[\mathrm{pa}_i], a) : \frac{1}{n} \sum_{i=1}^{N} \|(P_i - \hat{P}_{h,i})(\cdot | s_h^\tau[\mathrm{pa}_i], a_h^\tau)\|_1^2 \leq \xi_i, \forall i \in [d] \right\}.$$

where $\xi_i > 0$ are tuning parameters controlling the size of $\widehat{\mathcal{P}}_h$. We let $\texttt{ModelEst}(\mathbb{D}, \mathcal{P}_{\mathrm{M}}) = \{\widehat{\mathcal{P}}_h\}_{h=1}^H$.

Example II: $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP

Corollary: suboptimality of D^{2}MPO for $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP

Under Assumptions 1, 2, by proper choosing the tuning parameter $\{\xi_i\}_{i\in[d]}$, the suboptimality of D^{2}MPO for $\mathcal{S} \times \mathcal{A}$-rectangular factored tabular RMDP is

- when D is the TV-distance,

$$\mathrm{SubOpt}(\hat{\pi}; s_1) \leq \sqrt{C^{\star}_{\mathbb{P}^\star,\boldsymbol{\Phi}}} H^2 \cdot \sqrt{\frac{dC_1' \sum_{i=1}^{d} |\mathcal{O}|^{1+|\mathrm{pa}_i|} |\mathcal{A}| \log(C_2' N d/\delta)}{N}},$$

- when D is the KL-divergence, by $\rho = \min_{i\in[d]} \rho_i$,

$$\mathrm{SubOpt}(\hat{\pi}; s_1) \leq \frac{\sqrt{C^{\star}_{\mathbb{P}^\star,\boldsymbol{\Phi}}} H^2 \exp(H)}{\rho_{\min}} \cdot \sqrt{\frac{dC_1' \sum_{i=1}^{d} |\mathcal{O}|^{1+|\mathrm{pa}_i|} |\mathcal{A}| \log(C_2' N d/\delta)}{N}}.$$