Offline Guarded Safe Reinforcement Learning for Medical Treatment Optimization Strategies

Runze Yan[1*], Xun Shen[2*], Akifumi Wachi[3], Sebastien Gros[4], Anni Zhao[1], Xiao Hu[1]

1. Center for Data Science, Nell Hodgson Woodruff School of Nursing, Emory University
2. Department of Electrical Engineering and Computer Science, Tokyo University of Agriculture and Technology
3. LY Corporation
4. Department of Engineering Cybernetics, Norwegian University of Science and Technology

* Equal contribution

Why Offline Safe RL? Motivation & Use Cases

Why do we need offline safe reinforcement learning?

- **Healthcare**

- **Autonomous Driving**

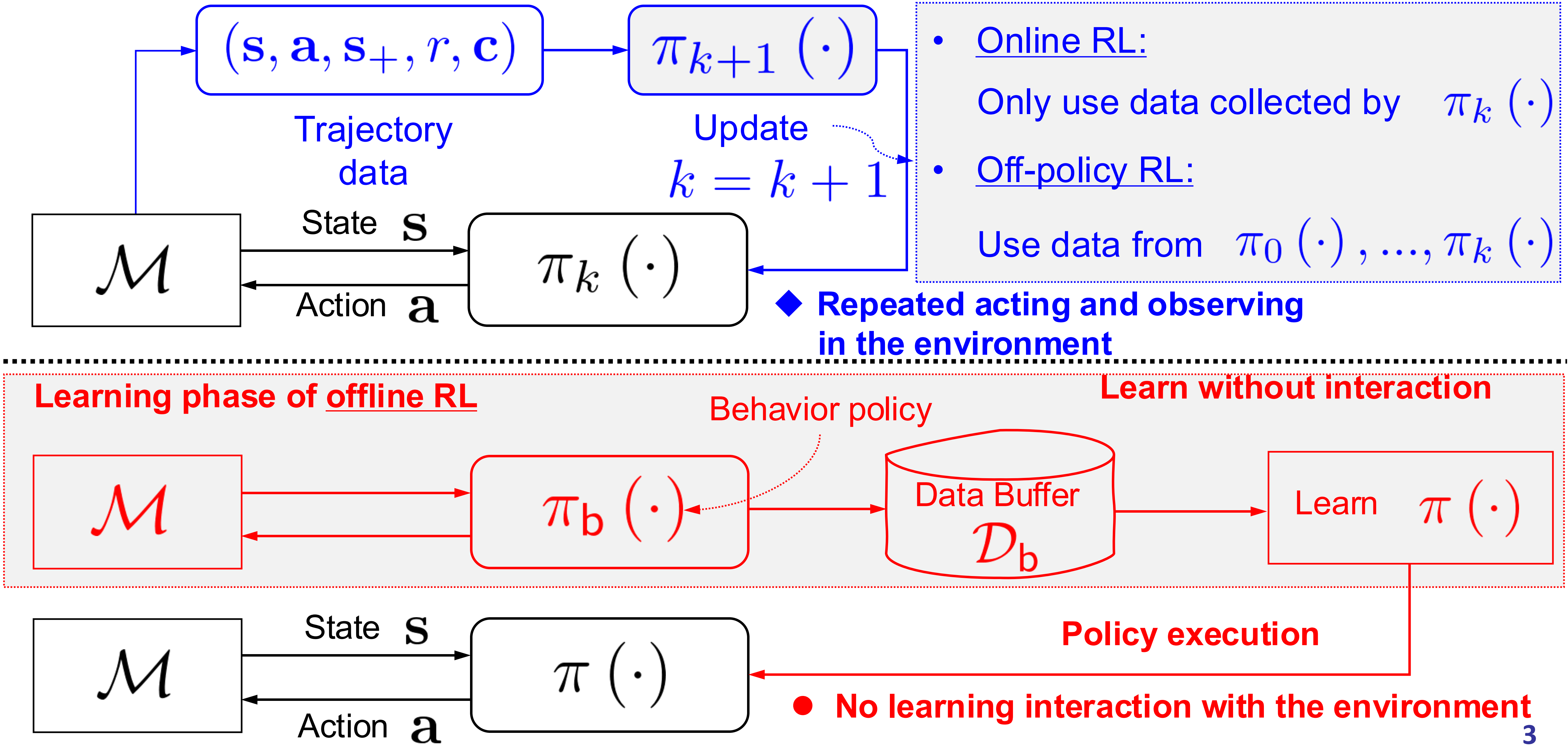
- **Chatbots**

Online exploration ⇨ risk of <u>unsafe behavior</u> ⇨ risk of <u>catastrophic failures</u>

- **Online exploration ❌. Offline policy learning with safety constraints**
- ✅ **Rich datasets obtained from <u>expert operations</u> (e.g., clinicians, drivers, etc.)**
- **Datasets collected under the <u>behavior policy</u> are available for use**

For practical deployment, offline safe reinforcement learning is indispensable.

How Offline RL Works: From Online Interaction to Logged Data



Core Challenge: Distribution Shift in Offline RL

Why offline safe RL difficult ? → **Prone to distribution shift**

Problem Setting

$$\max_{\theta\in\Theta} \quad V^{\theta}_{r,\mathcal{T}}(\rho_0)$$

$$\text{s.t.} \quad V^{\theta}_{c_j,\mathcal{T}}(\rho_0) \le \bar{c}_j, \quad \forall j \in [\ell].$$

- Dataset $\mathcal{D}_{\mathrm{b}} := \{(\mathbf{s}, \mathbf{a}, \mathbf{s}_+, r, \mathbf{c})\}$ ⇦ logged by the behavior policy π_{b}
- Policy Parametrization：$\pi \to \pi_\theta,\ \theta \in \Theta$
- Value function：

$$V^{\theta}_{\diamond,\mathcal{T}}(\rho_0) := \mathbb{E}_{\mathbf{s}\sim\rho_0}\left[\mathbb{E}_{\pi_\theta}\left[\sum_{h=0}^{\infty} \gamma^h \diamond(\mathbf{s}_h, \mathbf{a}_h) \mid \mathbf{s}_0 = \mathbf{s}\right]\right]$$

Ex：Model-free policy/value learning

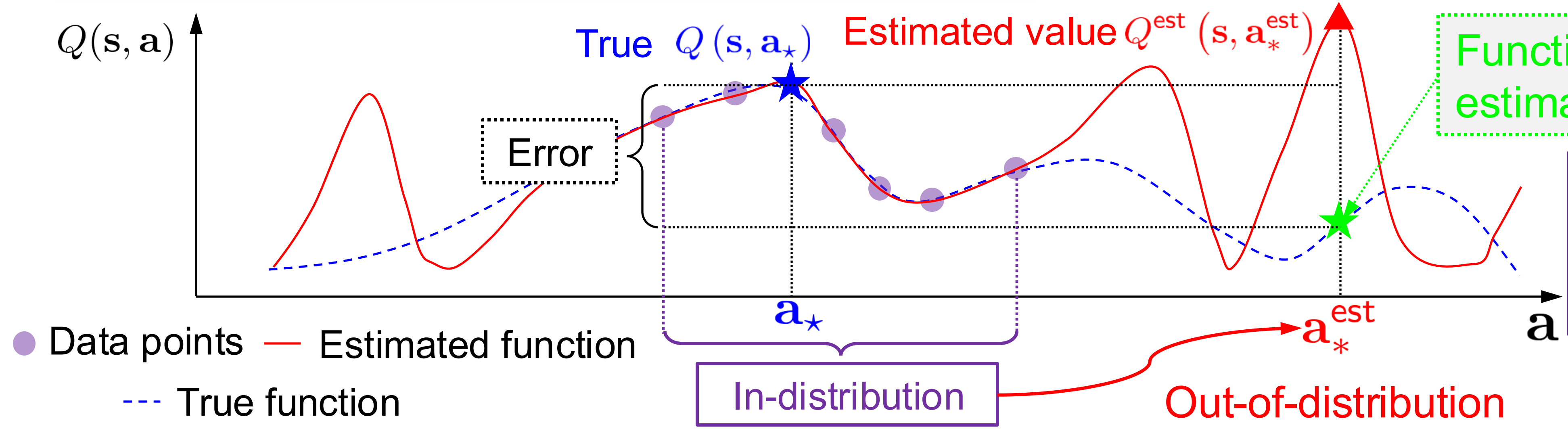
a. $\mathcal{D}_{\mathrm{b}} := \{(\mathbf{s}, \mathbf{a}, \mathbf{s}_+, r, \mathbf{c})\}$ ⇦Initial labeling$(\mathbf{s}, \mathbf{a}, Q^{\mathrm{est}}(\mathbf{s}, \mathbf{a}))$

b. Learn the value function

$$\min_{\vartheta} \quad \frac{1}{N}\sum_{i=1}^{N} \|Q^{\vartheta}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)}) - Q^{\mathrm{est}}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)})\|^2$$

c. Update via the model-free Bellman equation

$$Q^{\mathrm{est}}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)}) = r^{(i)} + \max_{\mathbf{a}_+} \ Q^{\vartheta}(\mathbf{s}_+^{(i)}, \mathbf{a}_+)$$



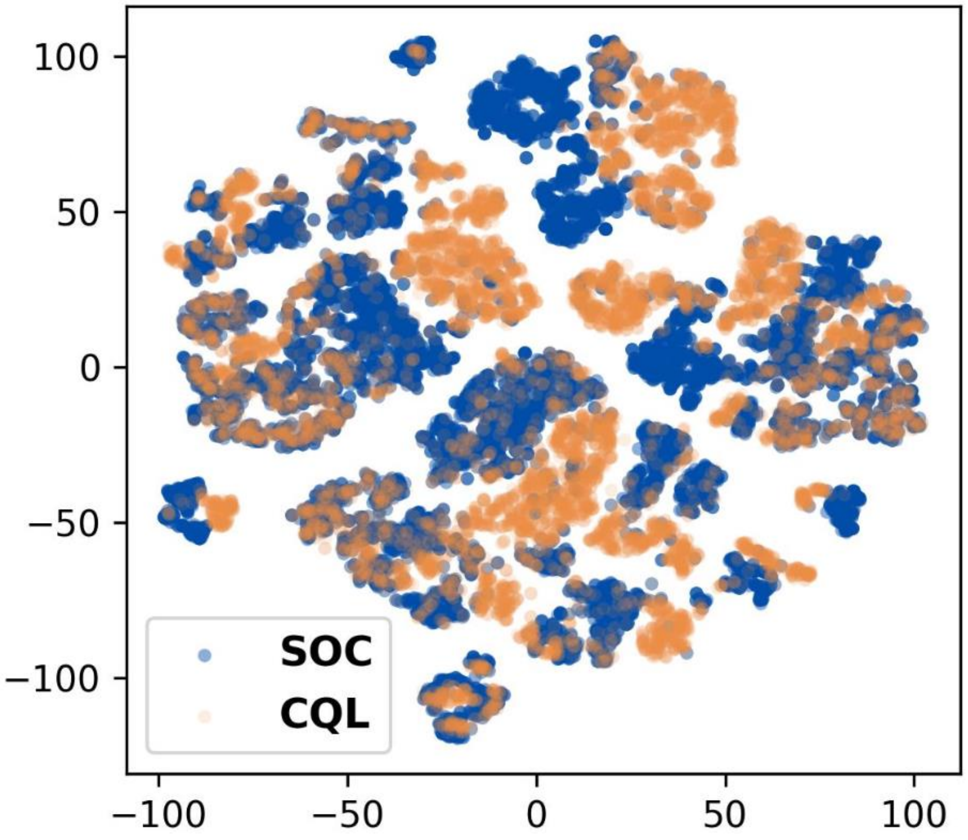
Ideas such CQL/BCQ：**penalize out-of-distribution actions, or imitate the logged behavior**

However……

State OOD Risk: Successor Occupancy & In-Distribution Trajectories

Even if we suppress action distribution shift, state distribution shift remains unsolved!

Even when an action isn't out-of-distribution, <u>its next state often is ODD</u>



Reflects the dataset's state support

SOC: successor $\mathbf{s}_{+}^{\text{soc}}$ after state $\mathbf{s}$ and behavior action $\mathbf{a}^{\text{soc}}$

Reflects the learned policy's state support

CQL: successor $\mathbf{s}_{+}^{\text{cql}}$ after state $\mathbf{s}$ and policy's action $\mathbf{a}^{\text{cql}}$

Action-value function $Q(\boxed{\mathbf{s}}, \mathbf{a})$

If the next state enters the OOD region $\rightarrow Q^{\text{est}}(\cdot)$ **becomes unreliable**

★ Avoid trajectories entering the OOD region → How?

★Solution from control engineering(<u>optimize state trajectories while staying in-distribution</u>)

- **Learn the OOD region by learning <u>a probabilistic control-invariant set</u>**
- **Impose a chance constraint on <u>trajectories entering the OOD region</u> (akin to chance-constrained MPC)**

Our Approach—Guardian: Guarded E-CMDP with Chance Constraints

Leverage control to **optimize state trajectories while keeping the policy within trustworthy, safe regions**

Data Buffer $\mathcal{D}_{\mathrm{b}}$ → Learn → Guardian's classifier / Dynamics model → Construct → Guarded E-CMDP / Guardian's constraint → π_{out}

Constrained Policy Optimization → Guarded E-CMDP

$$\widehat{\mathcal{U}}_{\theta,d} := \{\mathbf{x} \in \mathcal{U} : q(\mathbf{x}, \theta) \leq 1\}.$$

$$\hat{g}(\mathbf{s}, \mathbf{a}) = \mathbb{I}\left\{(\mathbf{s}, \mathbf{a}) \notin \widehat{\mathcal{U}}_{\theta,d}\right\}$$

Guarded Estimated Constrained MDP

$$\max_{\pi \in \Pi} \quad V^{\pi}_{\hat{r},\widehat{\mathcal{T}}}(\rho_0)$$

Guardian's constraint

$$\text{s.t.} \quad V^{\pi}_{\hat{g},\widehat{\mathcal{T}}}(\rho_0) \leq \bar{c}_{\hat{g}},$$

Safety Cost Value Function

$$V^{\pi}_{\hat{c}_j,\widehat{\mathcal{T}}}(\rho_0) \leq \bar{c}_j, \quad \forall j \in [\ell].$$

Theoretical guarantees of **safety, near-optimality, and in-distribution** preservation for the learned policy

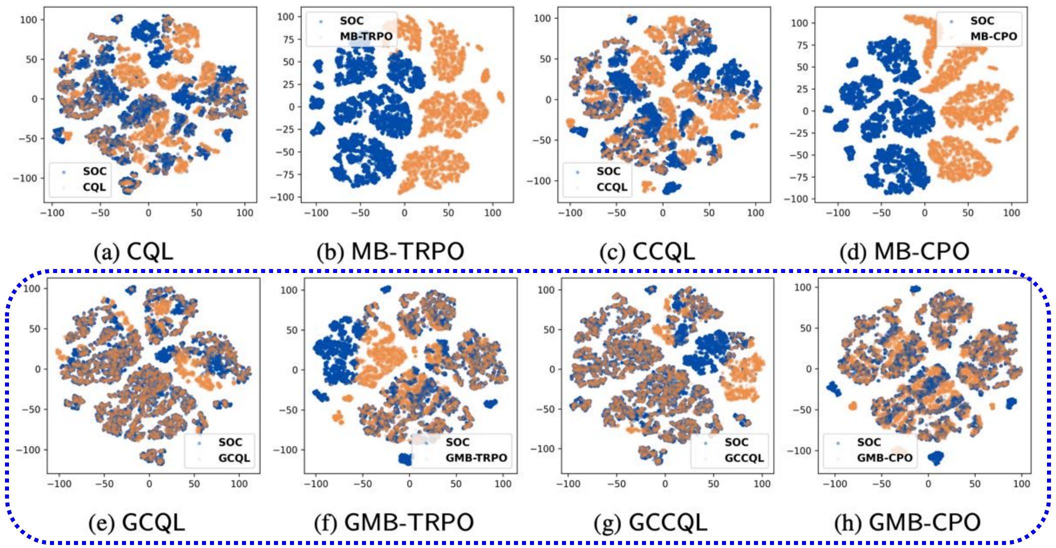
★Solution from control engineering

- Guardian's Classifier Learning the OOD region：learn **a probabilistic control-invariant set** ➡ Guardian's classifier
- Guardian's constrain impose **a probabilistic constraint on trajectories entering the OOD region** ➡ Guardian's constraint
- Impose **a probabilistic constraint on trajectories leaving the safe region**➡ Safety Cost Value Function

MIMIC-III Results: Safer States and Better Reward Outcomes

Validating on the real-world clinical dataset MIMIC-III

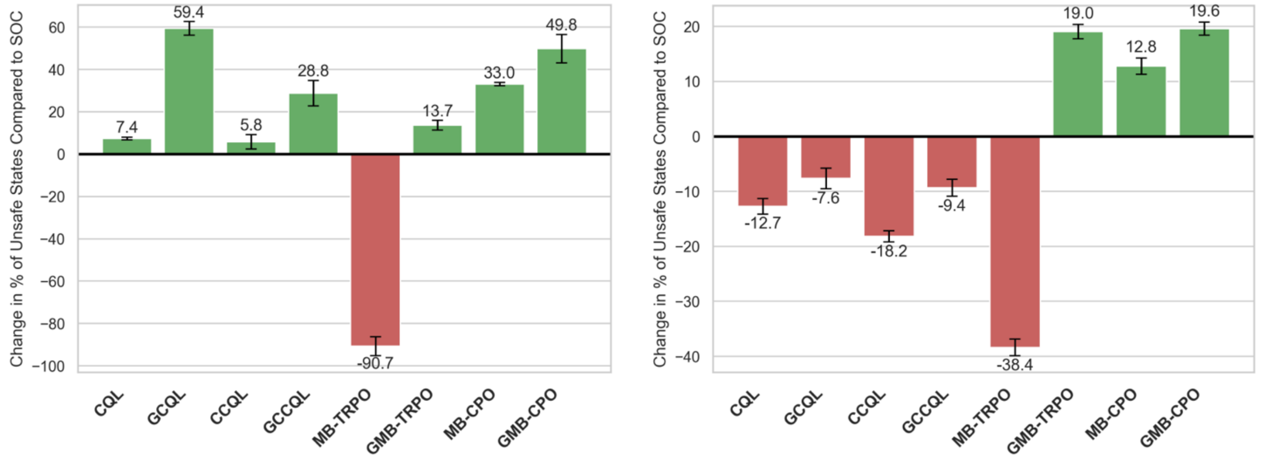
State distribution shift：Improved by Guardian



(a) CQL (b) MB-TRPO (c) CCQL (d) MB-CPO

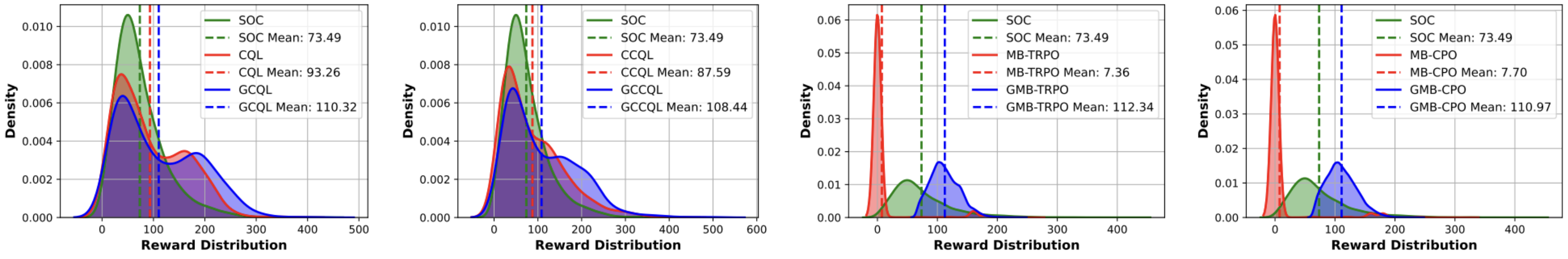
(e) GCQL (f) GMB-TRPO (g) GCCQL (h) GMB-CPO

State safety constraints：Improved by Guardian + state constrains



(a) SpO_2 (b) Urine Output

Reward outcomes：Improved by Guardian（means and distribution shape）



With the power of control theory

- Improves the reliability of ML methods (e.g., CQL)
- Clear enhances the reliability and safety of learning-augmented control