Efficient Safe Meta-Reinforcement Learning: Provable Near-Optimality and Anytime Safety

Siyuan Xu & Minghui Zhu

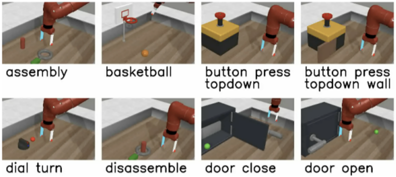
School of Electrical Engineering and Computer Science
The Pennsylvania State University

Conference on Neural Information Processing Systems
Dec, 2025

Meta-RL and Safe meta-RL

Meta-RL (or safe meta-RL):

Train a meta policy π_ϕ (meta-traning) such that π_ϕ can be adapted to a new RL (or safe RL) task τ_{new} by collecting a small dataset $\boldsymbol{D_t}$ of the task τ_{new} (meta-test).



Meta-RL v.s. Safe meta-RL (Constrained MDP definition)

The goal of adaptation (meta-test) in meta-RL:

$$\max_{\pi\in\Pi} J_{\tau_{new}}(\pi)$$

The goal of adaptation (meta-test) in safe meta-RL:

$$\max_{\pi\in\Pi} J_{\tau_{new}}(\pi)$$
$$\text{s.t. } J_{c_i,\tau_{new}}(\pi) \le d_i,\ \forall i = 1,\cdots,p$$

Higher requirement for policy adaptation in safe meta-RL

During meta-test time, we require safety-compliant policies for both exploration and deployment on the new task.

Anytime safety property: All the policies used to sample data (for policy adaptation) should satisfy the safety constraints of the new task τ_{new}.

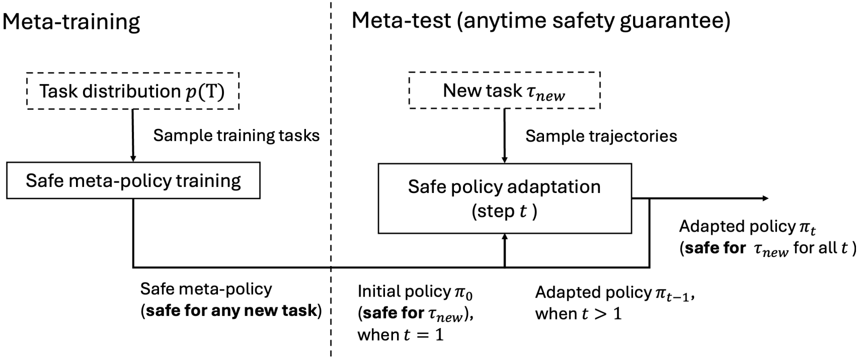
Data $\boldsymbol{D_t}$ for policy adaptation to new task τ_{new} in meta-RL:

- Data point (s_t, a_t, s_{t+1}, r_t)
- Any policy is feasible for the sampling of data $\boldsymbol{D_t}$

Data $\boldsymbol{D_t}$ for policy adaptation to new task τ_{new} in safe meta-RL:

- Data point $(s_t, a_t, s_{t+1}, r_t, c_t)$
- Policy used to sample data $\boldsymbol{D_t}$ is expected to be safe for the new task τ_{new}

Safe meta-RL framework



Overview:

- Meta-training: train a safe meta-policy π_ϕ from the task distribution
- Meta-test: take the meta-policy π_ϕ as the initial policy to iteratively adapt the policy to the new task τ_{new} by the safe policy adaptation

Safe policy adaptation

One safe policy adaptation step from π_ϕ

$$\pi^\tau = \mathcal{A}^s(\pi_\phi, \Lambda, \Delta, \tau) \triangleq \underset{\pi \in \Pi}{\operatorname{argmax}} \, \mathbb{E}_{s \sim \nu_\tau^{\pi_\phi}, a \sim \pi(\cdot|s)} \left[A_\tau^{\pi_\phi}(s,a)\right] - \lambda \, \mathbb{E}_{s \sim \nu_\tau^{\pi_\phi}} \left[D_{KL}\left(\pi(\cdot|s) \| \pi_\phi(\cdot|s)\right)\right],$$

$$\text{s.t. } J_{c_i,\tau}(\pi_\phi) + \mathbb{E}_{\substack{s \sim \nu_\tau^{\pi_\phi} \\ a \sim \pi(\cdot|s)}} \left[\frac{A_{c_i,\tau}^{\pi_\phi}(s,a)}{1-\gamma}\right] + \lambda_{c_i} \, \mathbb{E}_{s \sim \nu_\tau^{\pi_\phi}} \left[D_{KL}\left(\pi(\cdot|s) \| \pi_\phi(\cdot|s)\right)\right] \leq d_{i,\tau} + \delta_{c_i}.$$

When the parameter λ is properly selected and the initial policy π_ϕ satisfy the safety constraint:

- Solution existence: the feasibility set of the problem is not empty
- Safe policy guaranteed: the policy π^τ is safe for task τ, i.e., $J_{c_i,\tau}(\pi^\tau) \leq d_{i,\tau}, \forall i = 1, \cdots, p$.
- Monotonic improvement: the performance of π^τ is better than the meta-policy π_ϕ, i.e., $J_\tau(\pi^\tau) \geq J_\tau(\pi_\phi)$.

Safe meta-policy training

The optimization problem of the meta-policy training:

$$\max_{\phi} \mathbb{E}_{\tau \sim \mathbb{P}(\Gamma)}[J_\tau(\mathcal{A}^s(\pi_\phi, \Lambda, \Delta, \tau))],$$

$$\text{s.t. } J_{c_i,\tau}(\pi_\phi) \leq d_{i,\tau} + \delta_{c_i}, \forall i = 1, \cdots, p \text{ and } \forall \tau \in \Gamma.$$

- Objective design: the objective function is defined by the expected accumulated reward of the policy adapted from the meta-policy π_ϕ.
- Constraint design: the meta-policy π_ϕ satisfies the safety constraint for all tasks in the task distribution.
- Anytime safety achieved: all the adapted policies π_τ^1, π_τ^2, $\cdots$, π_τ^t, $\cdots$ are safe.

Closed-form solution for safe policy adaptation

Safe policy adaptation

$$\pi^{\tau} = \mathcal{A}^{s}(\pi_{\phi}, \Lambda, \Delta, \tau) \triangleq \underset{\pi \in \Pi}{\operatorname{argmax}} \, \mathbb{E}_{s \sim \nu_{\tau}^{\pi_{\phi}}, a \sim \pi(\cdot|s)} \left[A_{\tau}^{\pi_{\phi}}(s,a) \right] - \lambda \, \mathbb{E}_{s \sim \nu_{\tau}^{\pi_{\phi}}} \left[D_{KL}\left(\pi(\cdot|s) \| \pi_{\phi}(\cdot|s) \right) \right],$$

$$\text{s.t. } J_{c_i,\tau}(\pi_{\phi}) + \mathbb{E}_{\substack{s \sim \nu_{\tau}^{\pi_{\phi}} \\ a \sim \pi(\cdot|s)}} \left[\frac{A_{c_i,\tau}^{\pi_{\phi}}(s,a)}{1-\gamma} \right] + \lambda_{c_i} \, \mathbb{E}_{s \sim \nu_{\tau}^{\pi_{\phi}}} \left[D_{KL}\left(\pi(\cdot|s) \| \pi_{\phi}(\cdot|s) \right) \right] \leq d_{i,\tau} + \delta_{c_i}.$$

Under certain mild constraint qualifications, there exists Lagrangian multipliers $\{u_{c_i,\tau}^{*}\}_{i=1}^{p}$ with $0 \leq u_{c_i,\tau}^{*} < \infty$, such that

$$\pi^{\tau}(\cdot|s) \propto \exp(f_{\phi}(s,\cdot) + \eta^{-1}(A_{\tau}^{\pi_{\phi}}(s,\cdot) - \textstyle\sum_{i=1}^{p} u_{c_i,\tau}^{*} A_{c_i,\tau}^{\pi_{\phi}}(s,\cdot))),$$

for any $s \in \mathcal{S}$, where $\eta \triangleq \lambda + (1-\gamma) \sum_{i=1}^{p} u_{c_i,\tau}^{*} \lambda_{c_i}$.

- It is very efficient to solve the Lagrangian multipliers $\{u_{c_i,\tau}^{*}\}_{i=1}^{p}$ by the dual method.

Theoretical guarantee

Near-optimality and safety guarantee

Let $\lambda = \frac{2\gamma\alpha A^{max}}{1-\gamma}$, $\lambda_{c_i} = \frac{2\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2}$ and $\delta_{c_i} = \frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma)) - \epsilon$ for all $i = 1, \cdots, p$, where ϵ is chosen from $\left[0, \frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma))\right]$. Let ϕ^* be the solution of the meta-policy optimization problem. The solution of $\mathcal{A}^s(\pi_{\phi^*}, \Lambda, \Delta, \tau)$ exists, and we have

$$\mathbb{E}_{\tau\sim\mathbb{P}(\Gamma)}[J_\tau(\mathcal{A}^s(\pi_{\phi^*}, \Lambda, \Delta, \tau))] \geq \mathbb{E}_{\tau\sim\mathbb{P}(\Gamma)}[J_\tau(\pi^\tau_{*,[\epsilon]})] - \frac{4\gamma\alpha A^{max}}{(1-\gamma)^2} \mathcal{V}ar^\epsilon(\mathbb{P}(\Gamma)),$$

$$J_{c_i,\tau}(\mathcal{A}^s(\pi_{\phi^*}, \Lambda, \Delta, \tau)) - d_{i,\tau} \leq \frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma)) - \epsilon, \text{ for any } \tau \in \Gamma.$$

where $\pi^\tau_{*,[\epsilon]}$ is the ϵ-conservatively optimal policy defined as $\pi^\tau_{*,[\epsilon]} \triangleq \text{argmax}_{\pi\in\Pi} \; J_\tau(\pi) \; \text{ s.t. } \; J_{c_i,\tau}(\pi) \leq d_{i,\tau} - \epsilon$.

Theoretical guarantee

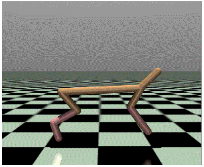
Case 1: Safety guaranteed

When $\delta_{c_i} = 0$, the safe constraint is strictly satisfied, i.e., $J_{c_i,\tau}(\pi^\tau) - d_{i,\tau} \leq 0$ for any τ, but the optimality comparator $J_\tau(\pi^\tau_{*,[\epsilon]})$ with $\epsilon = \frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma))$ is conservatively optimal (ϵ-conservatively optimal).

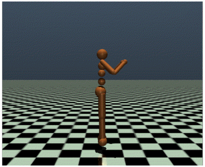
Case 2: Near-optimality

When $\delta_{c_i} = \frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma))$, the optimality comparator $J_\tau(\pi^\tau_{*,[0]})$ is the optimum, but the constraint is violated at most $\frac{4\gamma\alpha A^{max}_{c_i}}{(1-\gamma)^2} R(\mathbb{P}(\Gamma))$.

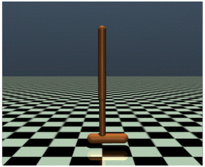
Experiments setting



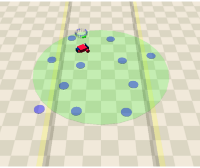
(a) Half-Cheetah



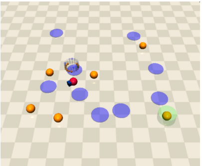
(b) Humanoid



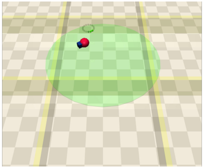
(c) Hopper



(c) Car-Circle



(d) Point-Button



(e) Point-Circle

Figure: Visualization of robotic locomotion environments, including Half-Cheetah, Humanoid, and Hopper, and collision avoidance tasks, including Car-Circle, Point-Button, and Point-Circle.

Experiments results

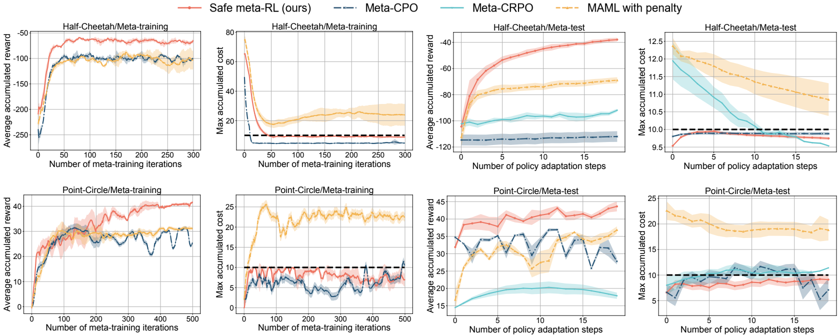


Figure: Average accumulated reward and maximal accumulated cost across all validation/test tasks during the meta-training and the meta-test in Half-Cheetah and Point-Circle. The accumulated reward and cost during meta-training are computed on the policy adapted one step from the meta-policy. The black dashed line is the constraint of the accumulated cost (below the line means satisfaction).

Experiments results

Computation time comparison:

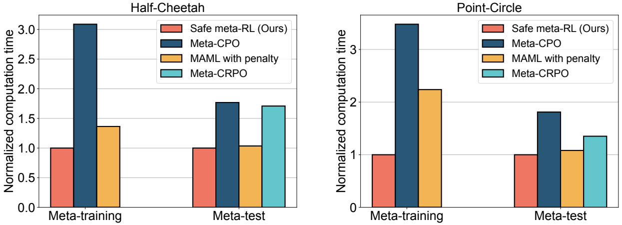


Figure: Normalized computation time of the meta-training (per iteration) and meta-test.

Conclusion

- Propose an efficient safe meta-RL framework.
- Theoretically guarantee the anytime safety and the near-optimality
- Experimentally validate the effectiveness of the algorithm in continuous control environments on locomotion tasks and collision avoidance tasks.