Policy Gradient Method For Robust Reinforcement Learning

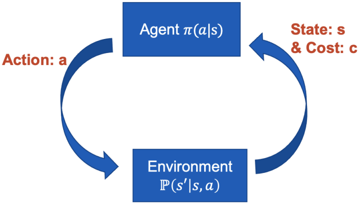
Yue Wang, **Shaofeng Zou**

University at Buffalo (SUNY)

July 2022

Reinforcement Learning (RL)

- An agent interacts with a stochastic environment: Markov Decision Process (MDP)
- MDP $(\mathcal{S}, \mathcal{A}, \mathsf{P}, c, \gamma)$
  - $\mathcal{S}$: state space
  - $\mathcal{A}$: action space
  - P: transition kernel
  - c: cost function
  - γ: discount factor



- A policy $\pi(a|s)$ is a conditional distribution over $\mathcal{A}$

Reinforcement Learning

- Value function for policy π at state s:

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)|S_0 = s, \pi\right]$$

- Goal: find an optimal policy that minimizes value function

$$\min_{\pi}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)|S_0 = s, \pi\right]$$

Robust RL

Training environment $\neq$ test environment

$\Rightarrow$ Model mismatch

$\Rightarrow$ Severe performance degradation

- modeling error between simulator and real-world applications
- non-stationary environment
- unexpected perturbations and potential adversarial attacks

Robust RL:

Find good policy that performs well under model mismatch

Robust MDP

- Robust MDP: $(\mathcal{S}, \mathcal{A}, \mathcal{P}, c, \gamma)$
  - $\mathcal{P}$: uncertainty set of transition kernels
  - Transition kernel at each time step comes from $\mathcal{P}$: $\kappa = (\mathsf{P}_0, \mathsf{P}_1, ...) \in \bigotimes_{t \geq 0} \mathcal{P}$
- Pessimistic approach in face of uncertainty[1]:
  - Robust value function:

$$V^\pi(s) = \max_{\kappa \in \bigotimes_{t \geq 0} \mathcal{P}} \mathbb{E}_\kappa \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) | S_0 = s, \pi \right]$$

  - Worst-case overall cost over uncertainty set
- Goal: Optimize the **worst-case** performance:

$$\min_\pi J_\rho(\pi) \triangleq \mathbb{E}_\rho[V^\pi(S)]$$

[1] Our results can be easily adapted to optimistic approach

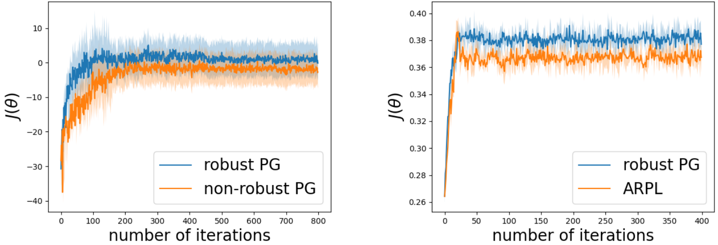
Related Works

- **Model-based robust MDP**: e.g., (Iyengar, 2005; Nilim and El Ghaoui, 2004; Bagnell et al., 2001; Satia and Lave Jr, 1973; Wiesemann et al., 2013; Tamar et al., 2014).
  Assume knowledge of uncertainty set and solve using dynamic programming, model-based and not scalable
- **Model-free value-based method**: e.g., (Roy et al., 2017; Badrinath and Kalathil, 2021; Wang and Zou, 2021).
  Value-based method, not scalable
- **Adversarial training approach for robust RL**: e.g., (Vinitsky et al., 2020; Pinto et al., 2017; Abdullah et al., 2019; Hou et al., 2020; Rajeswaran et al., 2017; Huang et al., 2017; Kos and Song, 2017; Pattanaik et al., 2018; Mandlekar et al., 2017).
  Empirical success but lack theoretical understanding

Main Contributions

- Derivation of robust policy gradient: $\partial V^{\pi_\theta}(s)$
- Global optimality guarantee and finite-time complexity bound
- Model-free robust actor-critic, its convergence and sample complexity

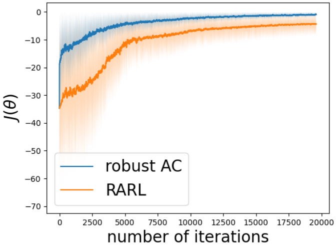
- Robust policy gradient v.s. vanilla policy gradient and ARPL Mandlekar et al. (2017)
- ARPL: Adversary randomly perturb observation then run vanilla policy gradient method using these perturbed samples
- Training on an unperturbed MDP, and evaluation on the worst-case transition kernel in $\mathcal{P}$

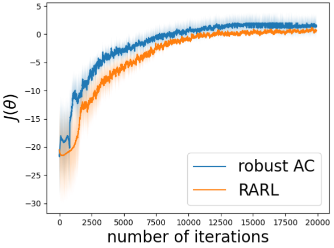


- Our robust policy gradient achieves higher reward on the worst-case transition kernel

Experiments

- Robust actor-critic v.s. RARL (Pinto et al., 2017)
- RARL: Adversary perturbs state transition. Agent and adversary are updated alternatively using gradient descent ascent.
- Training on an unperturbed MDP, and evaluation on the worst-case transition kernel in $\mathcal{P}$





- Our robust actor critic achieves higher reward on the worst-case transition kernel

R-Contamination Uncertainty Set (Huber, 1965)

- R-contamination: for some $0 \leq R \leq 1$,

$$\mathcal{P}_s^a = \left\{(1-R)p_s^a + Rq | q \in \Delta_{|\mathcal{S}|}\right\}$$

  - p_s^a is "centroid" of $\mathcal{P}_s^a$, which is *unknown*
  - R is the design parameter of the uncertainty set, which measures the size of the uncertainty set
- (s, a)-rectangular uncertainty set: $\mathcal{P} = \bigotimes_{s,a} \mathcal{P}_s^a$
- Motivation: systems suffering from random perturbations, and adversarial attacks at each time step

Robust Policy Gradient

- Idea: derive gradient of $J_\rho(\pi) \triangleq \mathbb{E}_\rho[V^\pi(S)]$, and run gradient descent
- Robust value function V^π is not differentiable everywhere because of max over κ

$$V^\pi(s) = \max_\kappa \mathbb{E}_\kappa \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) | S_0 = s, \pi \right]$$

- Major challenge lies in the max operator

Robust Policy Sub-gradient

- Consider a parametric policy class $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta\}$

Theorem (Robust Policy Sub-gradient)

Define

$$\psi_\rho(\theta) \triangleq \frac{\gamma R}{(1-\gamma)(1-\gamma+\gamma R)} \sum_{s \in \mathcal{S}} d_{s_\theta}^{\pi_\theta}(s) \sum_{a \in \mathcal{A}} \nabla \pi_\theta(a|s) Q^{\pi_\theta}(s,a) + \frac{1}{1-\gamma+\gamma R} \sum_{s \in \mathcal{S}} d_\rho^{\pi_\theta}(s) \sum_{a \in \mathcal{A}} \nabla \pi_\theta(a|s) Q^{\pi_\theta}(s,a),$$

then (1) almost everywhere in Θ, $J_\rho(\theta)$ is differentiable and
$\psi_\rho(\theta) = \nabla J_\rho(\theta)$*;*
(2) at non-differentiable θ, $\psi_\rho(\theta) \in \partial J_\rho(\theta)$.

- $\partial J_\rho(\theta)$: set of Fréchet sub-differential (Kruger, 2003) of J_ρ at θ
- Reduces to vanilla policy gradient if $R = 0$

Robust Policy Sub-gradient Algorithm

Input: T, α_t
Initialization: θ_0
FOR $t = 0, 1, ..., T-1$
$\theta_{t+1} \leftarrow \prod_{\Theta}(\theta_t - \alpha_t \psi_\mu(\theta_t))$
Output: θ

- Vanilla policy gradient is able to find globally optimal policy for non-robust RL, e.g., (Bhandari and Russo, 2021; Agarwal et al., 2021; Cen et al., 2021)
- Question: is robust policy sub-gradient able to converge to global optimum of $J_\rho(\theta)$?
- Answer: Yes!

Convex-Like: PL-Condition

PL-condition (Karimi et al., 2016; Bolte et al., 2007):

Theorem (PL-Condition)

Under direct policy parameterization,

$$J_\rho(\theta) - J_\rho^* \leq C_{PL} \max_{\hat{\pi} \in (\Delta(\mathcal{A}))^{|\mathcal{S}|}} \left\langle \pi_\theta - \hat{\pi}, \psi_\rho(\theta) \right\rangle .$$

Robust Policy Sub-gradient: Global Optimality

Theorem (Global Optimality under Direct Parameterization)

If $\alpha_t > 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$, then under direct policy parameterization, θ_T converges to a global optimum of $J_\rho(\theta)$ as $T \to \infty$ almost surely.

- Sub-gradient method converges to stationary points: $\{\theta : 0 \in \partial J_\rho(\theta)\}$
- Stationary point is globally optimal due to PL-condition

Derivation of Robust Policy Sub-gradient

- Basic idea:
  - Recursively apply robust Bellman equation:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}(V^\pi) \right)$$

$$= \sum_{a \in \mathcal{A}} \pi(a|s) \left(c(s,a) + \gamma(1-R) \sum_{s' \in \mathcal{S}} p^a_{s,s'} V^\pi(s') + R \max_{s'} V^\pi(s') \right)$$

(under R-contamination uncertainty set)

  - and use the fact: $\partial(f) + \partial(g) \subseteq \partial(f+g)$

Smoothed Robust Policy Gradient

Robust policy sub-gradient method:

- Complexity is generally difficult to establish

> Solution: smoothed robust policy gradient

Smoothed Robust Policy Gradient

Smoothed robust Bellman operator:

$$\mathrm{T}_\sigma^\pi V(s) = \mathbb{E}_{A\sim\pi(\cdot|s)}\left[c(s,A) + \gamma(1-R)\sum_{s'\in\mathcal{S}} p_{s,s'}^A V(s') + \gamma R \cdot \mathrm{LSE}(\sigma, V)\right],$$

where $\mathrm{LSE}(\sigma, V) = \frac{\log(\sum_{i=1}^d e^{\sigma V(i)})}{\sigma}$ for $V \in \mathbb{R}^d$ and some $\sigma > 0$

- $\mathrm{LSE}(\sigma, V)$ converges to $\max_s V(s)$ as $\sigma \to \infty$
- T_σ^π is a contraction, V_σ^π is the fixed point of T_σ^π
  softmax will not induce contraction (Asadi and Littman, 2017)
- V_σ^π is differentiable in θ and converges to V^π as $\sigma \to \infty$

Smoothed Robust Policy Gradient

- $J_\rho^\sigma(\theta) = \sum_{s\in\mathcal{S}} \rho(s) V_\sigma^{\pi_\theta}(s)$: smoothed robust objective
- Gradient of $J_\rho^\sigma(\theta)$:

$$\nabla J_\rho^\sigma(\theta) = B(\rho,\theta) + \frac{\gamma R \sum_{s\in\mathcal{S}} e^{\sigma V_\sigma^{\pi_\theta}(s)} B(s,\theta)}{(1-\gamma)\sum_{s\in\mathcal{S}} e^{\sigma V_\sigma^{\pi_\theta}(s)}},$$

where $B(s,\theta) \triangleq \frac{1}{1-\gamma+\gamma R} \sum_{s'\in\mathcal{S}} d_s^\pi(s') \sum_{a\in\mathcal{A}} \nabla \pi_\theta(a|s') Q_\sigma^{\pi_\theta}(s',a)$, and $B(\rho,\theta) \triangleq \mathbb{E}_{S\sim\rho}[B(S,\theta)]$.

- Smoothed robust policy gradient: $\theta_{t+1} \leftarrow \prod_\Theta(\theta_t - \alpha_t \nabla J_\rho^\sigma(\theta))$

Even though gradient is for J_ρ^σ, the algorithm can still find a global optimum of J_ρ by choosing a large σ

Global optimality and Complexity

Consider direct policy parameterization

Theorem

For any $\epsilon > 0$, set $\sigma = \mathcal{O}(\epsilon^{-1})$ and $T = \mathcal{O}(\epsilon^{-3})$, then

$$\min_{t \leq T-1} J(\theta_t) - J^* \leq 3\epsilon.$$

- If $R = 0$, i.e., no robustness is considered, complexity reduces to $\mathcal{O}(\epsilon^{-2})$, which matches with vanilla policy gradient in (Agarwal et al., 2021)

Model-free Robust Actor-Critic

- Recall robust policy subgradient:

$$\psi_\rho(\theta) \triangleq \frac{\gamma R}{(1-\gamma)(1-\gamma+\gamma R)} \sum_{s\in\mathcal{S}} d^{\pi_\theta}_{s_\theta}(s) \sum_{a\in\mathcal{A}} \nabla \pi_\theta(a|s) Q^{\pi_\theta}(s,a)$$
$$+ \frac{1}{1-\gamma+\gamma R} \sum_{s\in\mathcal{S}} d^{\pi_\theta}_{\rho}(s) \sum_{a\in\mathcal{A}} \nabla \pi_\theta(a|s) Q^{\pi_\theta}(s,a)$$

- $Q^{\pi_\theta}(s,a)$ measures cost under worst-case transition kernel and π_θ, however, only samples from simulator are available

Monte Carlo does not work

Critic: Robust TD

- Parametric robust action value function Q_ζ, e.g., linear function approximation or neural network.

Input: T_c, π, β_t
Initialization: ζ, s_0
Choose $a_0 \sim \pi(\cdot|s_0)$
FOR $t = 0, 1, ..., T_c - 1$
Observe c_t, s_{t+1}
Choose $a_{t+1} \sim \pi(\cdot|s_{t+1})$
$V_t^* \leftarrow \max_s \left\{\sum_{a\in\mathcal{A}} \pi(a|s) Q_\zeta(s,a)\right\}$
$\delta_t \leftarrow Q_\zeta(s_t, a_t) - \underbrace{\left(c_t + \gamma(1-R)Q_\zeta(s_{t+1}, a_{t+1}) + \gamma R V_t^*\right)}_{\text{robust target}}$ (robust TD error)
$\zeta \leftarrow \zeta - \beta_t \delta_t \nabla_\zeta Q_\zeta(s_t, a_t)$
Output: ζ

Robust Actor-Critic Algorithm

- Using robust TD algorithm to estimate robust Q-function in (smoothed) robust policy gradient
- Under tabular setting, global optimality can be established, overall sample complexity is $\mathcal{O}(\epsilon^{-7})$

Robust actor-critic algorithm can be applied with arbitrary value function/policy approximation.

Summary

- Robust policy gradient with provable global optimality
- Model-free robust actor-critic algorithm
- Can be easily scaled to large/continuous problems
- Future direction: model-free robust RL algorithms for uncertainty sets defined by, e.g., KL divergence, Wasserstein distance

https://arxiv.org/abs/2205.07344, ICML Baltimore 2022

Questions?

Reference I

Abdullah, M. A., Ren, H., Ammar, H. B., Milenkovic, V., Luo, R., Zhang, M., and Wang, J. (2019). Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2021). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76.

Asadi, K. and Littman, M. L. (2017). An alternative softmax operator for reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 243–252. JMLR.

Badrinath, K. P. and Kalathil, D. (2021). Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *Proc. International Conference on Machine Learning (ICML)*, pages 511–520. PMLR.

Bagnell, J. A., Ng, A. Y., and Schneider, J. G. (2001). Solving uncertain Markov decision processes.

Reference II

Bhandari, J. and Russo, D. (2021). On the linear convergence of policy gradient methods for finite MDPs. In *Proc. International Conference on Artifical Intelligence and Statistics (AISTATS)*, pages 2386–2394. PMLR.

Bolte, J., Daniilidis, A., and Lewis, A. (2007). The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2021). Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*.

Hou, L., Pang, L., Hong, X., Lan, Y., Ma, Z., and Yin, D. (2020). Robust reinforcement learning with Wasserstein constraint. *arXiv preprint arXiv:2006.00945*.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. (2017). Adversarial attacks on neural network policies. In *Proc. International Conference on Learning Representations (ICLR)*.

Reference III

Huber, P. J. (1965). A robust version of the probability ratio test. *Ann. Math. Statist.*, 36:1753–1758.

Iyengar, G. N. (2005). Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280.

Karimi, H., Nutini, J., and Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer.

Kos, J. and Song, D. (2017). Delving into adversarial attacks on deep policies. In *Proc. International Conference on Learning Representations (ICLR)*.

Kruger, A. Y. (2003). On Fréchet subdifferentials. *Journal of Mathematical Sciences*, 116(3):3325–3358.

Reference IV

Mandlekar, A., Zhu, Y., Garg, A., Fei-Fei, L., and Savarese, S. (2017). Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939. IEEE.

Nilim, A. and El Ghaoui, L. (2004). Robustness in Markov decision problems with uncertain transition matrices. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 839–846.

Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., and Chowdhary, G. (2018). Robust deep reinforcement learning with adversarial attacks. In *Proc. International Conference on Autonomous Agents and MultiAgent Systems*, pages 2040–2042.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. (2017). Robust adversarial reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, pages 2817–2826. PMLR.

Reference V

Rajeswaran, A., Ghotra, S., Ravindran, B., and Levine, S. (2017). Epopt: Learning robust neural network policies using model ensembles. In *Proc. International Conference on Learning Representations (ICLR)*.

Roy, A., Xu, H., and Pokutta, S. (2017). Reinforcement learning under model mismatch. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 3046–3055.

Satia, J. K. and Lave Jr, R. E. (1973). Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740.

Tamar, A., Mannor, S., and Xu, H. (2014). Scaling up robust MDPs using function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 181–189. PMLR.

Vinitsky, E., Du, Y., Parvate, K., Jang, K., Abbeel, P., and Bayen, A. (2020). Robust reinforcement learning using adversarial populations. *arXiv preprint arXiv:2008.01825*.

Reference VI

Wang, Y. and Zou, S. (2021). Online robust reinforcement learning with model uncertainty. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*.

Wiesemann, W., Kuhn, D., and Rustem, B. (2013). Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183.

Derivation of Robust Sub-gradient

Assume there exists $\phi(\theta) \in \partial(\max_s V^{\pi_\theta}(s))$, and note that $\partial(f) + \partial(g) \subseteq \partial(f+g)$, hence from robust Bellman equation,

$$
\begin{aligned}
&\partial V^{\pi_\theta}(s)\\
&\supseteq \sum_{a\in\mathcal{A}} \nabla\pi_\theta(a|s) Q^{\pi_\theta}(s,a)\\
&\quad + \sum_{a\in\mathcal{A}} \pi_\theta(a|s)\partial\left(c(s,a) + \gamma(1-R)\sum_{s'\in\mathcal{S}} p^a_{s,s'} V^{\pi_\theta}(s') + \gamma R \max_s V^{\pi_\theta}(s)\right)\\
&\supseteq \sum_{a\in\mathcal{A}} \nabla\pi_\theta(a|s) Q^{\pi_\theta}(s,a) + \sum_{a\in\mathcal{A}} \pi_\theta(a|s)\left(\gamma(1-R)\sum_{s'\in\mathcal{S}} p^a_{s,s'}\partial V^{\pi_\theta}(s')\right) + \gamma R\phi(\theta)\\
&= \gamma R\phi(\theta) + \sum_{a\in\mathcal{A}} \nabla\pi_\theta(a|s) Q^{\pi_\theta}(s,a) + \gamma(1-R)\sum_{s'\in\mathcal{S}} \mathbb{P}(S_1 = s'|S_0 = s, \pi_\theta)\partial V^{\pi_\theta}(s')
\end{aligned}
$$

Derivation of Robust Sub-gradient

Recursively applying the above,

$$\partial \max_s V^{\pi_\theta}(s) \ni \frac{\gamma R}{1-\gamma+\gamma R}\phi(\theta) + \frac{1}{1-\gamma+\gamma R}\sum_{s'\in\mathcal{S}} d^{\pi_\theta}_{s_\theta}(s')\sum_{a\in\mathcal{A}}\nabla\pi_\theta(a|s')Q^{\pi_\theta}(s',a)$$

- $d^\pi_s(s') \triangleq (1-\gamma+\gamma R)\sum_{t=0}^\infty \gamma^t(1-R)^t\cdot\mathbb{P}(S_t=s'|S_0=s,\pi)$: discounted visitation distribution
- $s_\theta \triangleq \arg\max_s V^{\pi_\theta}(s)$: worst state under π_θ

Hence for at differentiable θ, $\partial \max_s V^{\pi_\theta}(s) = \{\nabla \max_s V^{\pi_\theta}(s)\} = \{\phi(\theta)\}$.
Thus

$$\phi(\theta) = \frac{\gamma R}{1-\gamma+\gamma R}\phi(\theta) + \frac{1}{1-\gamma+\gamma R}\sum_{s'\in\mathcal{S}} d^{\pi_\theta}_{s_\theta}(s')\sum_{a\in\mathcal{A}}\nabla\pi_\theta(a|s')Q^{\pi_\theta}(s',a),$$

and $\phi(\theta)$ can be solved.
It can be proved that $\phi(\theta)$ is a sub-differential of $\max_s V^{\pi_\theta}(s)$ at any θ by verifying the definition of sub-differential

Algorithm 4 Robust Actor-Critic

Input: T, T_c, σ, α_t, M
Initialization: θ_0
for $t = 0, 1, ..., T-1$ **do**
  Run Algorithm 3 for T_c times
  $Q_t \leftarrow Q_{\zeta_{T_c}}$
  $V_t(s) \leftarrow \sum_{a\in\mathcal{A}} \pi_\theta(a|s)Q_t(s,a)$ for all $s \in \mathcal{S}$
  for $j = 1, ..., M$ **do**
    Sample $T^j \sim$ **Geom**$(1-\gamma+\gamma R)$
    Sample $s_0^j \sim \rho$
    Sample trajectory from s_0^j: $(s_0^j, a_0^j, ..., s_{T^j}^j)$ following π_{θ_t}
    $B_t^j \leftarrow \frac{1}{1-\gamma+\gamma R} \sum_{a\in\mathcal{A}} \nabla\pi_\theta(a|s_{T^j}^j)Q_t(s_{T^j}^j, a)$
    $x_0^j \leftarrow \arg\max_s V_t(s)$
    Sample trajectory from x_0^j: $(x_0^j, b_0^j, ..., x_{T^j}^j)$ following π_{θ_t}
    $D_t^j \leftarrow \frac{1}{1-\gamma+\gamma R} \sum_{a\in\mathcal{A}} \nabla\pi_\theta(a|x_{T^j}^j)Q_t(x_{T^j}^j, a)$
    $g_t^j \leftarrow B_t^j + \frac{\gamma R}{1-\gamma} D_t^j$
  end for
  $g_t \leftarrow \frac{\sum_{j=1}^M g_t^j}{M}$
  $\theta_{t+1} \leftarrow \prod_\Theta(\theta_t - \alpha_t g_t)$
end for
Output: θ_T