SELF-SUPERVISED CONTINUOUS CONTROL WITHOUT POLICY GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the remarkable progress made by the policy gradient algorithms in reinforcement learning (RL), sub-optimal policies usually result from the local exploration property of the policy gradient update. In this work, we propose a method called Zeroth-Order Supervised Policy Improvement (ZOSPI) that exploits the estimated value function Q globally while preserving the local exploitation of the policy gradient methods. Experiments show that ZOSPI achieves competitive results on the MuJoCo benchmarks with a remarkable sample efficiency. Moreover, different from the conventional policy gradient methods, the policy learning of ZOSPI is conducted in a self-supervised manner. We show such a self-supervised learning paradigm has the flexibility of including optimistic exploration as well as adopting a non-parametric policy.

1 INTRODUCTION

Model-free Reinforcement Learning has achieved great successes in many challenging tasks (Mnih et al., 2015; Vinyals et al., 2019; Pachocki et al.), however one obstacle for its application to real-world control problems is the insufficient sample efficiency. To improve the sample efficiency, off-policy methods (Degris et al., 2012; Gu et al., 2016; Wang et al., 2016; Lillicrap et al., 2015; Fujimoto et al., 2018) reuse the experiences generated by previous policies to optimize the current policy, therefore obtaining a higher sample efficiency than on-policy methods (Schulman et al., 2015; 2017). Alternatively, SAC (Haarnoja et al., 2018) increases sample efficiency by conducting more active exploration of current policy, which is achieved by applying a maximum entropy framework (Haarnoja et al., 2017) to the off-policy actor critic, and also results in the state-of-the-art asymptotic performance. OAC (Ciosek et al., 2019) further improves SAC by combining it with the Upper Confidence Bound heuristics (Brafman & Tennenholtz, 2002) to conduct more informative exploration. Despite of their improvements, these methods all rely on a Gaussianized policy and a local exploration strategy that simply adds noises to the action space, thus they might still lead to sub-optimal solutions as pointed out by Tessler et al. (2019).

In this work we aim to explore a new learning paradigm that is able to carry out non-local exploration as well as non-local exploitation in continuous control tasks and achieve higher sample efficiency. Specifically, to better exploit the learned value function Q, we propose to search it globally for a better action, in contrast to previous attempts in policy gradient methods that only utilize its local information (e.g. the Jacobian matrix used in (Silver et al., 2014)). The idea behind our work is mostly related to the value-based policy gradient methods (Lillicrap et al., 2015; Fujimoto et al., 2018), where the policy gradient optimization step takes the role of finding a well-performing action given a learned state-action value function. In previous works, the policy gradient step tackles the curse of dimensionality since it is intractable to directly search for the maximal value in the continuous action space (Mnih et al., 2015).

To perform a global exploitation of the learned value function Q, we propose to apply a zeroth-order optimization scheme and update the target policy through supervised learning, which is inspired by works of evolution strategies (Salimans et al., 2017; Conti et al., 2018; Mania et al., 2018) that adopt zeroth-order optimizations in the parameter space. Different from the standard policy gradient, combining the zeroth-order optimization with supervised learning forms a new way of policy update. Such a update avoids the local improvement of policy gradient: when policy gradient is applied, the target policy uses policy gradient to adjust its predictions according to the deterministic policy

gradient theorem (Silver et al., 2014), but such adjustments can only lead to local improvements and may induce sub-optimal policies due to the non-convexity of the policy function (Tessler et al., 2019); on the contrary, the combination of zeroth-order optimization and supervised learning proposed in this paper can help the non-convex policy optimization to escape the local minimum as shown in our experiments.

Our contributions are summarized as follows. We first introduce a simple yet novel policy optimization method, namely the Zeroth-Order Supervised Policy Improvement (ZOSPI), where the policy utilizes global information of the learned value function Q and adjusts itself through sample-based supervised learning. Then we show the capability of ZOSPI by comparing it with SOTA policy gradient methods on the MuJoCo locomotion benchmarks, demonstrating its remarkable sample efficiency as well as on-par final performance. Finally, we further demonstrate the flexibility of ZOSPI by exposing two potential extensions of ZOSPI, namely the combination of ZOSPI and optimistic explorations and the compatibility of ZOSPI with non-parametric policies such as Gaussian Processes policies.

2 Related Work

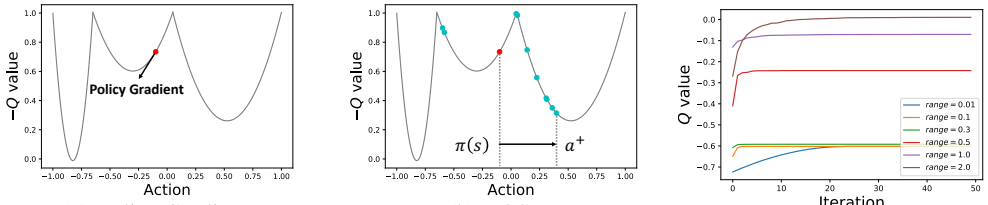
Policy Gradient Methods. The policy gradient methods solve the MDP by directly optimizing the policy to maximize the cumulative reward (Williams, 1992; Sutton & Barto, 1998). While the prominent on-policy policy gradient methods like TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) improve the learning stability via trust region updates, off-policy methods such as DPG (Silver et al., 2014) and DDPG (Lillicrap et al., 2015) can learn with a higher sample efficiency than on-policy methods. The work of TD3 (Fujimoto et al., 2018) further addresses the function approximation error and boost the stability of DDPG with several improvements. Another line of works is the combination of policy gradient methods and the max-entropy principle, which leads to better exploration and stable asymptotic performances (Haarnoja et al., 2017; 2018). All of these approaches adopt function approximators (Sutton et al., 2000) for state or state-action value estimation as well as directionally uninformed Gaussian policies for policy parameterization, which lead to a local exploration behavior (Ciosek et al., 2019; Tessler et al., 2019).

Self-Supervised RL. Self-supervised learning or self-imitate learning is a rising stream as an alternative approach for model-free RL. Instead of applying policy gradient for policy improvement, methods of self-supervised RL update policies through supervised learning by minimizing the mean square error between target actions and current actions predicted by a policy network (Sun et al., 2019), or alternatively by maximizing the likelihood for stochastic policy parameterizations (Ghosh et al., 2019). While these works focus on the Goal-Conditioned tasks, in this work we aim at general RL tasks. Some other works use supervised learning to optimize the policy towards manually-selected policies to achieve better training stability (Wang et al., 2018; Zhang et al., 2019; Abdolmaleki et al., 2018), but rather in this work our our policy optimization is purely based on self-supervision with a much simpler formulation while achieving efficient performance.

Zeroth-Order Methods. Zeroth-order optimization methods, also called gradient-free methods, are widely used when gradients are difficult to compute. They approximate the local gradient with random samples around the current estimate. The works in Wang et al. (2017); Golovin et al. (2019) show that a local zeroth-order optimization method has a convergence rate that depends logarithmically on the ambient dimension of the problem under some sparsity assumptions. It can also efficiently escape saddle points in non-convex optimizations (Vlatakis-Gkaragkounis et al., 2019; Bai et al., 2020). In RL, many studies have verified an improved sample efficiency of zeroth-order optimization (Usunier et al., 2016; Mania et al., 2018; Salimans et al., 2017). In this work we provide a novel way of combining the local sampling and the global sampling to ensure that our algorithm approximates the gradient descent locally and is also able to find a better global region.

3 Preliminaries

We consider the deterministic Markov Decision Process (MDP) with continuous state and action spaces in the discounted infinite-horizon setting. Such MDPs can be denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are continuous, and the unknown state transition probability representing the transition dynamics is denoted by $P : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$. $r : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the reward function and $\gamma \in [0, 1]$ is the discount factor. An MDP $\mathcal{M}$ and a learning



(a) Policy Gradient (b) ZOSPI (c) Simulation

Figure 1: (a) Q value landscape of a 1-dim continuous control task. Policy gradient methods optimize the policy according to the local information of Q (b) For the same task, ZOSPI directly updates the predicted actions to the sampled action with the largest Q value. (c) Simulation results, where in each optimization iteration 10 actions are uniformly sampled with different ranges, and reported results are averaged over 100 random seeds. It can be seen that a larger random sample range improves the chance of finding global optima. Similar phenomenon also exist in practice as shown in Appendix A.

algorithm operating on $\mathcal{M}$ with an arbitrary initial state $s_0 \in \mathcal{S}$ constitute a stochastic process described sequentially by the state s_t visited at time step t, the action a_t chosen by the algorithm at step t, the reward $r_t = r(s_t, a_t)$ and the next state $s_{t+1} = P(s_t, a_t)$ for any $t = 0, \ldots, T$. Let $H_t = \{s_0, a_0, r_0, \ldots, s_t, a_t, r_t\}$ be the trajectory up to time t. Our algorithm finds the policy that maximizes the discounted cumulative rewards $\sum_{t=0}^{T} \gamma^t r_t$. Our work follows the general Actor-Critic framework (Konda & Tsitsiklis, 2000; Peters & Schaal, 2008; Degris et al., 2012; Wang et al., 2016), which learns in an unknown environment using a value network denoted by $Q_{w_t} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ for estimating Q values and a policy network for learning the behavior policy $\pi_{\theta_t} : \mathcal{S} \mapsto \mathcal{A}$. Here w_t and θ_t are respectively the parameters of these two networks at step t.

4 Zeroth-Order Self-Supervised Continuous Control

4.1 A Motivating Example

Figure 1 shows a motivating example to demonstrate the benefits of applying zeroth-order optimization to policy updates. Consider we have learned a Q function that has multiple local optima[1], and our present deterministic policy selects a certain action at this state, denoted as the red dot in Figure 1(a). In deterministic policy gradient methods (Silver et al., 2014; Lillicrap et al., 2015), the policy gradient is conducted according to the chain rule to update the policy parameter θ with regard to Q-value by timing up the Jacobian matrix $\nabla_\theta \pi_\theta(s)$ and the derivative of Q, *i.e.* $\nabla_a Q(s, a)$. Consequently, the policy gradient can only lead to a local improvement, and similar local improvement behaviors are also observed in stochastic policy gradient methods like PPO and SAC (Schulman et al., 2017; Haarnoja et al., 2018; Tessler et al., 2019; Ciosek et al., 2019). Instead, if we are able to sample sufficient *random* actions in a broader range of the action space, denoted as blue dots in Figure 1(b), and then evaluate their values respectively through the learned Q estimator, it is possible to find the action with a higher Q value as the target action in the optimization. Figure 1(c) shows the simulation results using different sample ranges for the sample-based optimization starting from the red point. It is clear that a larger sample range improves the chance of finding the global optima. Utilizing such a global exploitation on the learned value function is the key insight of this work.

4.2 Zeroth-Order Supervised Policy Improvement

Q-learning in the tabular setting relies on finding the best action given the current state, which can be difficult in the continuous action space due to the non-convexity of Q. Instead, a policy network is thus trained to approximate the solution. In most of previous policy gradient methods, the policy class is selected to be Gaussian in consideration of both exploration and computational tractability,

[1]Here we assume the traditional estimation of Q function is sufficient for a global exploitation (Fujimoto et al., 2018; Haarnoja et al., 2018) and we will discuss an improved estimation method in the next section.

Algorithm 1 Zeroth-Order Policy Optimization

Require
Objective function Q_s, domain $\mathcal{A}$, current point $a_0 = \pi_\theta(s)$, number of local samples n_1 and global samples n_2, local scale $\eta > 0$ and step size h.
Locally sampling
Sample n points around a_0 by

$$a_i = a_0 + \mu e_i \text{ for } e_i \sim \mathcal{N}(a_0, I_d), i = 1, \dots n_1,$$

where $\mathcal{N}(a_0, I_d)$ is the standard normal distribution centered at a_0.
Globally sampling
Sample n points uniformly in the entire space by

$$a_{i+n_1} \sim \mathcal{U}_\mathcal{A}, \text{ for } i = 1, \dots, n_2,$$

where $\mathcal{U}_\mathcal{A}$ is the uniform distribution over $\mathcal{A}$.
Update
Set $a^+ = \arg\max_{a \in \{a_0, \dots a_{n_1+n_2}\}} Q_s(a)$.
Update policy π_θ according to Eq.(2)

while, in this work, we consider the deterministic policy class which is simpler and easier to learn as presented in Silver et al. (2014).

As shown in Silver et al. (2014), DPG updates the policy network only through the first-order gradient of the current Q value estimation:

$$\nabla_a Q_{w_t}(s_t, \pi_{\theta_t}(s_t)) \nabla_\theta \pi_{\theta_t}(s_t). \tag{1}$$

Such an update through the local information of Q may incur a slow convergence rate, especially when Q function is non-convex. To mitigate this issue, we propose the Zeroth-Order Supervised Policy Improvement (ZOSPI), which exploits the entire learned Q function instead of merely the local information of Q. Thus the key insight of our proposed method is to utilize the zeroth-order method to overcome the local policy improvement problem induced by the non-convexity of Q.

To be specific, we first calculate the predicted action $a_0 = \pi_{\theta_t}(s_t)$. Then we sample two sets of actions with size n, namely a local set and a global set. For the local set, we sample actions randomly from a Gaussian distribution centered at a_0. For the global set, we sample points uniformly over the action space. The update a_t^+ is chosen as the action that gives the highest Q value in the union of two sets. Finally, we apply the supervised policy improvement that minimizes the L_2 distance between a_t^+ and $\pi_{\theta_t}(s_t)$, which gives the descent direction:

$$\nabla_\theta \frac{1}{2}(a_t^+ - \pi_{\theta_t}(s_t))^2 = (a_t^+ - \pi_{\theta_t}(s_t)) \nabla_\theta \pi_{\theta_t}(s_t). \tag{2}$$

The implementation detail is shown in Algorithm 1.

Comparison between two methods. We now compare the performances obtained via applying Eq.(2) to the standard deterministic policy gradient update used in Eq.(1).

Proportion 1. Following the definition in Algorithm 1, let the best action in the local set be $a_L = \arg\max_{a_i, i=1,\dots n} Q_{w_t}(s_t, a_i)$. We have $a_L - a_0 \propto \nabla_a Q_{w_t}(s_t, a_0)$, when $n \to \infty$ and $\eta \to 0$.

Proportion 1, derived directly from the definition of gradient, guarantees that with a sufficiently large number of samples and a sufficiently small η, zeroth-order optimization can at least find the local descent direction as in a first-order method.

When the best action is actually included in the global set, zeroth-order optimization will update towards the global direction with a larger step size as $\mathbb{E}\|a_1 - \pi_{\theta_t}(s_t)\| \leq \mathbb{E}\|a_{n+1} - \pi_{\theta_t}(s_t)\|$ in general. *The benefits of ZOSPI over DPG are determined by the probability of sampling an action globally that is close to the global minima*. We will discuss this with more details in Appendix A.1. With a sufficient number of sampled actions at each step, ZOSPI is able to find better solutions in terms of higher Q values for a given state, and therefore can globally exploit the Q function, which is especially useful when the Q function is non-convex as illustrated in Figure 1.

Algorithm 2 Zeroth-Order Supervised Policy Improvement (ZOSPI)

Require

- Number of epochs M, size of mini-batch N, momentum $\tau > 0$.
- Random initialized policy network π_{θ_1}, target policy network $\pi_{\theta'_1}$, $\theta'_1 \leftarrow \theta_1$.
- Two random initialized Q networks, and corresponding target networks, parameterized by $w_{1,1}, w_{2,1}, w'_{1,1}, w'_{2,1}$. $w'_{i,1} \leftarrow w_{i,1}$.
- Empty experience replay buffer $\mathcal{D} = \{\}$.

for iteration $= 1, 2, ...$ **do**
  for t $= 1, 2, ..., T$ **do**
    # Interaction
    Run policy $\pi_{\theta'_t}$ in environment, store transition tuples (s_t, a_t, s_{t+1}, r_t) into $\mathcal{D}$.
    for epoch $= 1, 2, ..., M$ **do**
      Sample a mini-batch of transition tuples $\mathcal{D}' = \{(s_{t_j}, a_{t_j}, s_{t_j+1}, r_{t_j})\}_{j=1}^{N}$.
      # Update Q
      Calculate target Q value $y_j = r_{t_j} + \min_{i=1,2} Q_{w'_{i,t}}(s_{t_j+1}, \pi_{\theta'_t}(s_{t_j}))$.
      Update w_{it} with one step gradient descent on the loss $\sum_j (y_j - Q_{w'_{i,t}}(s_{t_j}, a_{t_j}))^2$, $i = 1, 2$.
      # Update π
      Call Algorithm 1 for policy optimization to update θ_t.
    end for
    $\theta'_{t+1} \leftarrow \tau\theta_t + (1-\tau)\theta'_t$.
    $w'_{i,t+1} \leftarrow \tau w_{i,t} + (1-\tau)w'_{i,t}$.
    $w_{i,t+1} \leftarrow w_{i,t}$; $\theta_{t+1} \leftarrow \theta_t$.
  end for
end for

Algorithm 2 provides the pseudo code for ZOSPI, where we follow the double Q network in TD3 as the critic, and also use target networks for stability. In the algorithm we sample actions in the global set from a uniform distribution on the action space $a_i \sim \mathcal{U}_{\mathcal{A}}$, and sample actions from an on-policy local Gaussian (e.g., $a_i(s) = \pi_{\theta_{old}}(s) + \eta_i$, and $\eta_i \sim \mathcal{N}(0, \sigma^2)$)) to form the local set and guarantee local exploitation, so that ZOSPI will at least perform as good as deterministic policy gradient methods (Silver et al., 2014; Lillicrap et al., 2015; Fujimoto et al., 2018).

5 Benefits of Continuous Control without Policy Gradient

Different from standard policy gradient methods, the policy optimization step in ZOSPI can be interpreted as sampling-based supervised learning. Such difference provides several potential benefits and enable further extensions of this work. Here we discuss the combination of ZOSPI with UCB exploration as well as non-parametric models in RL.

5.1 Better Exploration with Bootstrapped Networks

Sample efficient RL requires algorithms to balance exploration and exploitation. One of the most popular way to achieve this is called optimism in face of uncertainty (OFU) (Brafman & Tennenholtz, 2002; Jaksch et al., 2010; Azar et al., 2017; Jin et al., 2018), which gives an upper bound on Q estimates and applies the optimal action corresponding to the upper bound. The optimal action a_t is given by the following optimization problem:

$$\arg\max_a Q^+(s_t, a), \tag{3}$$

where Q^+ is the upper confidence bound on the optimal Q function. A guaranteed exploration performance requires both a good solution for (3) and a valid upper confidence bound.

While it is trivial to solve (3) in the tabular setting, the problem can be intractable in a continuous action space. Therefore, as shown in the previous section, ZOSPI adopts a local set to approximate policy gradient descent methods in the local region and further applies a global sampling scheme to increase the potential chance of finding a better maxima.

As for the requirement of a valid upper confidence bound, we use bootstrapped Q networks to address the uncertainty of Q estimates as in Osband et al. (2016; 2018); Agarwal et al. (2019); Kumar et al. (2019); Ciosek et al. (2019). Specifically, we keep K estimates of Q, namely $Q_1, \dots Q_K$ with bootstrapped samples from the replay buffer. Let $\overline{Q} = \frac{1}{K}\sum_k Q_k(s,a)$. An upper bound Q^+ is

$$Q^+(s,a) = \overline{Q} + \phi\sqrt{\frac{1}{K}\sum_k [Q_k(s,a) - \overline{Q}]^2}, \tag{4}$$

where ϕ is the hyper-parameter controlling the failure rate of the upper bound. Another issue is on the update of bootstrapped Q networks. Previous methods (Agarwal et al., 2019) usually update each Q network with the following target $r_t + \gamma Q_k\left(s_{t+1}, \pi_{\theta_t}(s_{t+1})\right)$, which violates the Bellman equation as π_{θ_t} is designed to be the optimal policy for Q^+ rather than Q_k. Using π_{θ_t} also introduces extra dependencies among the K estimates. We instead employ a global random sampling method to correct the violation as

$$r_t + \gamma \max_{i=1,\dots n} Q_k\left(s_{t+1}, a_i\right), \quad a_1, \dots a_n \sim \mathcal{U}_{\mathcal{A}}.$$

The correction also reinforces the argument that a global random sampling method yields a good approximation to the solution of the optimization problem (3). The detailed algorithm is provided in Algorithm 4 in Appendix C.

5.2 Leveraging Gaussian Processes in Continuous Control

Different from previous policy gradient methods, the self-supervised learning paradigm of ZOSPI permits it to learn both its actor and critic with a regression formulation. Such a property enables the learning of actor in ZOSPI to be implemented with either parametric models like neural networks or non-parametric models like Gaussian Processes (GP). Although plenty of previous works have discussed the application of GP in RL by virtue of its natural uncertainty capture ability, most of these works are limited to model-based methods or discrete action spaces for value estimation (Kuss & Rasmussen, 2004; Engel et al., 2005; Kuss, 2006; Levine et al., 2011; Grande et al., 2014; Fan et al., 2018). On the other hand, ZOSPI formulates the policy optimization in continuous control tasks as a regression objective, therefore empowers the usage of GP policy in continuous control tasks.

As a first attempt of applying GP policies in continuous control tasks, we simply alter the actor network with a GP to interact with the environment and collect data, while the value approximator is still parameterized by a neural network. We leave the investigation of better consolidation design in the future.

6 Experiments

In this section, we show the empirical results to demonstrate the effectiveness of our proposed ZOSPI method on the MuJoCo locomotion tasks, and provide diagnostic environment with known optimal policy to show the proposed extensions discussed in Sec. 5 . Specifically, we validate the following statements:

1. If we use ZOSPI with locally sampled actions, the performance of ZOSPI should be the same as its policy gradient counterpart (i.e., TD3); if we increase the sampling range, ZOSPI will be able to better exploit the Q function and find better solutions than the methods based on policy gradient.
2. If we continuously increase the sampling range, it will result in an uniform sampling (in practice we include an additional local sampling to encourage local improvements in the later stage of the learning process), and the Q function can be maximally exploited.
3. ZOSPI can be combined with Bootstrapped Q-value estimation and behave with the principle OFU Brafman & Tennenholtz (2002) to pursue better exploration, or combined with GP to lay a foundation of future works of continuous control with non-parametric models.

ZOSPI on the MuJoCo Locomotion Tasks. In this section we evaluate ZOSPI on the OpenAI Gym locomotion tasks based on the MuJoCo engine (Brockman et al., 2016; Todorov et al., 2012).

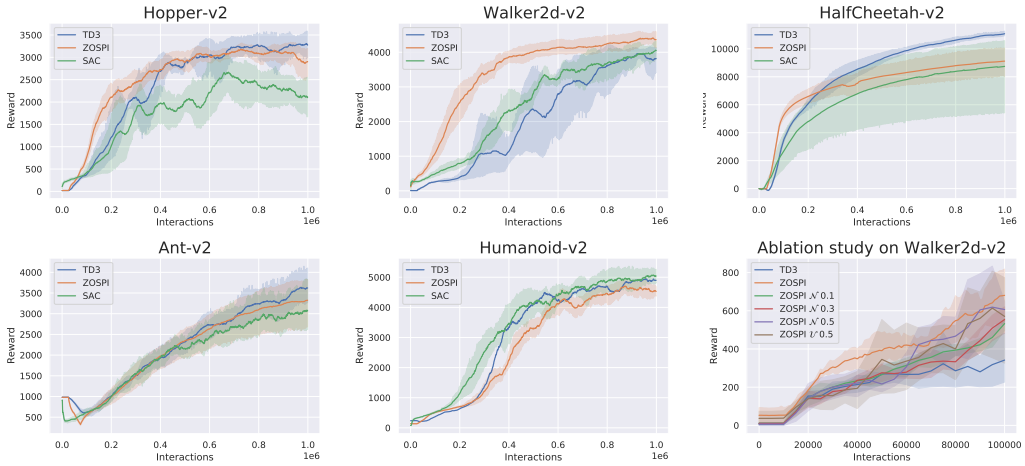


Figure 2: Experimental results on the MuJoCo locomotion tasks. The shaded region represents half a standard deviation of the average evaluation over 10 trials.

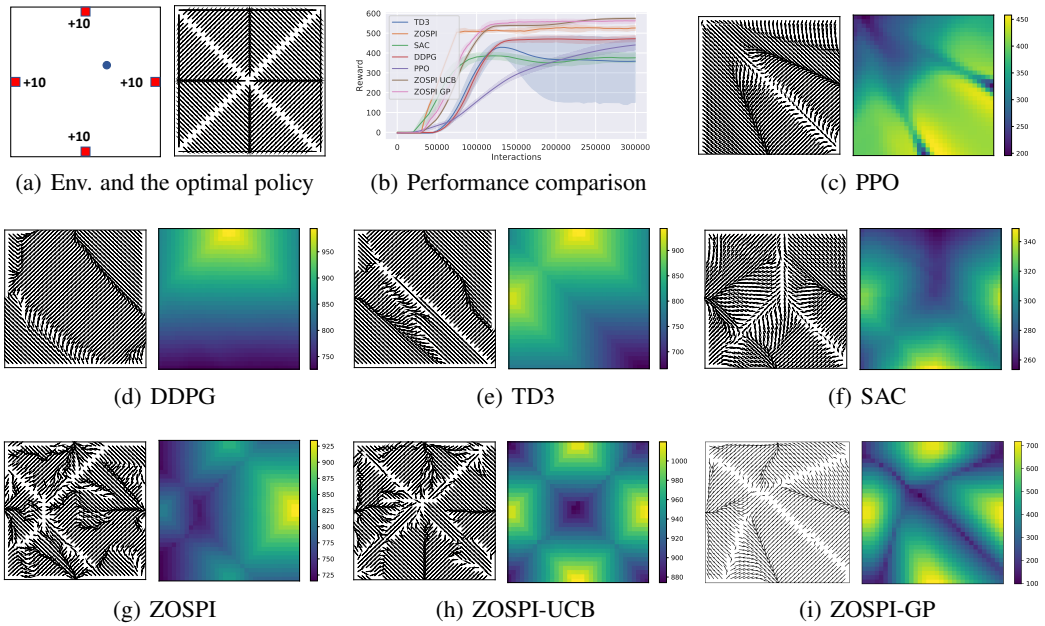
Concretely, we test ZOSPI on five locomotion environments, namely the Hopper-v2, Walker2d-v2, HalfCheetah-v2, Ant-v2 and Humanoid-v2 and include TD3 and SAC, respectively the deterministic and the stochastic SOTA policy gradient methods in the comparison. We compare results of different methods within 1×10^6 environment interactions to demonstrate the high learning efficiency of ZOSPI.The results of TD3 are obtained by running the code released by the author and the results of SAC are directly extracted from the training logs reported in (Haarnoja et al., 2018).

The results of ZOSPI, TD3, and SAC are included in Figure 2. It is worth noting that in our implementation of ZOSPI, only 50 actions are sampled for all tasks and it is sufficient to learn well-performing policies. Surprisingly, with such high sampling efficiencies, the results of ZOSPI are good even in the tasks that have high-dimensional action spaces such as Ant-v2 and Humanoid-v2. While a total of 50 sampled actions should be very sparse in the high dimensional space, we contribute the success of ZOSPI to the generality of the policy network as well as the sparsity of meaningful actions, *i.e.*, even in tasks that have high dimension action spaces, only limited dimensions of actions are crucial for making decisions.

The last plot in Figure 2 shows the ablation study on the sampling range $\mathcal{N}$ in ZOSPI, where a sampling method based on a zero-mean Gaussian is applied and we gradually increase its variance from 0.1 to 0.5. We also evaluate the uniform sampling method with radius of 0.5, which is denoted as $\mathcal{U}$ 0.5 in the Figure. The results suggest that zeroth-order optimization with local sampling performs similarly to the policy gradient method, and increasing the sampling range can effectively improve the performance.

Extensions on the Four-Solution-Maze. The Four-Solution-Maze (FSM) environment is a diagnostic environment where four positive reward regions with a unit side length are placed in the middle points of 4 edges of a $N \times N$ map. An agent starts from a uniformly initialized position in the map and can then move in the map by taking actions according to the location observations (current coordinates x and y). Valid actions are limited to $[-1, 1]$ for both x and y axes. Each game consists of $2N$ timesteps for the agent to navigate in the map and collect rewards. In each timestep, the agent will receive a $+10$ reward if it is inside one of the 4 reward regions or a tiny penalty otherwise. For simplicity, there are no obstacles in the map, the optimal policy thus will find the nearest reward region, directly move towards it, and stay in the region till the end. Figure 3(a) visualizes the environment and the ground-truth optimal solution.

On this environment we compare ZOSPI to on-policy and off-policy SOTA policy gradient methods in terms of the learning curves, each of which is averaged by 5 runs. The results are presented in Figure 3(b). And learned policies from different methods are visualized in Figure 3(c)-3(i). For each method we plot the predicted behaviors of its learned policy at grid points using arrows (although the environment is continuous in the state space), and show the corresponding value function of its



(a) Env. and the optimal policy (b) Performance comparison (c) PPO

(d) DDPG (e) TD3 (f) SAC

(g) ZOSPI (h) ZOSPI-UCB (i) ZOSPI-GP

Figure 3: Visualization of learned policies on the FSM environment. (a) the FSM environment and its optimal solution, where the policy should find the nearest reward region and move toward it; (b) learning curves of different approaches; (c)-(i) visualize the learned policies and corresponding value functions.

learned policy with a colored map. All policies and value functions are learned with 0.3M interactions except for SAC whose figures are learned with 1.2M interactions as it can find 3 out of 4 target regions when more interactions are provided.

We use 4 bootstrapped Q networks for the upper bound estimation in consideration of both better value estimation and computational cost for ZOSPI with UCB. And in ZOSPI with GP, a GP model is used to replace the actor network in data-collection, i.e.,exploration. The sample efficiency of ZOSPI is much higher than that of other methods. Noticeably ZOSPI with UCB exploration is the only method that can find the optimal solution, *i.e.*, a policy directs to the nearest region with a positive reward. All other methods get trapped in sub-optimal solutions by moving to only part of reward regions they find instead of moving toward the nearest one.

7 Conclusion

In this work, we propose the Zeroth-Order Supervised Policy Improvement (ZOSPI) method as an alternative approach to policy gradient methods for continuous control tasks. We evaluate ZOSPI on the MuJoCo locomotion tasks, where our method achieves competitive performance in terms of sample efficiency and final performance compared to SOTA policy gradient methods (TD3, SAC). Different from previous policy gradient methods, ZOSPI is based on self-supervised learning and the learning of its actor can be conducted with regression. Such a property enables several extensions such like optimistic exploration and non-parametric policies which can not be seamlessly deployed on policy gradient methods, therefore opens up potential future directions. On a diagnostic environment called Four-Solution-Maze, the proposed method is shown to outperform prevailing policy gradient methods in terms of both sample efficiency and final performance. Besides, ZOSPI with optimistic exploration is the only algorithm that is able to find the near-optimal solution.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Jonas Degrave, Steven Bohez, Yuval Tassa, Dan Belov, Nicolas Heess, and Martin Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*, 2019.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Qinbo Bai, Mridul Agarwal, and Vaneet Aggarwal. Escaping saddle points for zeroth-order nonconvex optimization using estimated gradient descent. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6. IEEE, 2020.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor critic. In *Advances in Neural Information Processing Systems*, pp. 1785–1796, 2019.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Advances in Neural Information Processing Systems*, pp. 5027–5038, 2018.

Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pp. 201–208, 2005.

Ying Fan, Letian Chen, and Yizhou Wang. Efficient model-free reinforcement learning using gaussian process. *arXiv preprint arXiv:1812.04359*, 2018.

Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Dibya Ghosh, Abhishek Gupta, Justin Fu, Ashwin Reddy, Coline Devine, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals without reinforcement learning. *arXiv preprint arXiv:1912.06088*, 2019.

Daniel Golovin, John Karro, Greg Kochanski, Chansoo Lee, Xingyou Song, et al. Gradientless descent: High-dimensional zeroth-order optimization. *arXiv preprint arXiv:1911.06317*, 2019.

Robert Grande, Thomas Walsh, and Jonathan How. Sample efficient reinforcement learning with gaussian processes. In *International Conference on Machine Learning*, pp. 1332–1340, 2014.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1352–1361. JMLR. org, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pp. 11761–11771, 2019.

Malte Kuss. *Gaussian process models for robust regression, classification, and reinforcement learning*. PhD thesis, echnische Universität Darmstadt Darmstadt, Germany, 2006.

Malte Kuss and Carl E Rasmussen. Gaussian processes in reinforcement learning. In *Advances in neural information processing systems*, pp. 751–758, 2004.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 19–27, 2011.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pp. 4026–4034, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8617–8629, 2018.

Jakub Pachocki, Greg Brockman, Jonathan Raiman, Susan Zhang, Henrique Pondé, Jie Tang, Filip Wolski, Christy Dennison, Rafal Jozefowicz, Przemyslaw Debiak, et al. Openai five, 2018. *URL https://blog. openai. com/openai-five*.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

Hao Sun, Zhizhong Li, Xiaotong Liu, Bolei Zhou, and Dahua Lin. Policy continuation with hindsight inverse dynamics. In *Advances in Neural Information Processing Systems*, pp. 10265–10275, 2019.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 1998.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Chen Tessler, Guy Tennenholtz, and Shie Mannor. Distributional policy optimization: An alternative approach for continuous control. *arXiv preprint arXiv:1905.09855*, 2019.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pp. 5026–5033. IEEE, 2012. ISBN 978-1-4673-1737-5. URL `http://dblp.uni-trier.de/db/conf/iros/iros2012.html#TodorovET12`.

Nicolas Usunier, Gabriel Synnaeve, Zeming Lin, and Soumith Chintala. Episodic exploration for deep deterministic policies for starcraft micromanagement. 2016.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, and Georgios Piliouras. Efficiently avoiding saddle points with zero order methods: No gradients required. In *Advances in Neural Information Processing Systems*, pp. 10066–10077, 2019.

Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. Exponentially weighted imitation learning for batched historical data. In *Advances in Neural Information Processing Systems*, pp. 6288–6297, 2018.

Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. *arXiv preprint arXiv:1710.10551*, 2017.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Chuheng Zhang, Yuanqi Li, and Jian Li. Policy search by target distribution learning for continuous control. *arXiv preprint arXiv:1905.11041*, 2019.

A VISUALIZATION OF Q-LANDSCAPE

Figure 4 shows the visualization of learned policies (actions given different states) and Q values in TD3 during training in the Pendulum-v0 environment, where the state space is 3-dim and action space is 1-dim. The red lines indicates the selected action by the current policy. The learned Q function are always non-convex, as a consequence, in many states the TD3 is not able to find globally optimal solution and locally gradient information may be misleading in finding actions with high Q values.

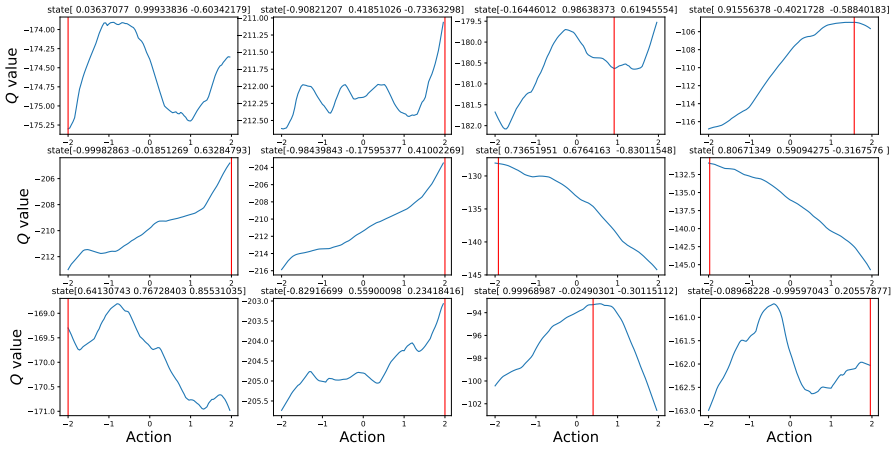


Figure 4: Landscape of learned value function in the Pendulum-v0 environment

A.1 DISCUSSION ON THE BENEFITS OF GLOBAL SAMPLING

In this section, we discuss a type of structure, with which our zeroth-order optimization has a exponential convergence rate. To better explain our points, we include an Algorithm 3 in Appendix B, a modified version of Algorithm 1, which prevents the sampled action jumping too far across different global regions.

Definition 1 (Sampling-Easy Functions)**.** A function $F : \mathcal{X} \subset \mathbb{R}^d \mapsto \mathbb{R}$ is called $\alpha\beta$-*Sampling-Easy*, if it has an unique global minima x^*, and there exists an region $\mathcal{D} \subset \mathcal{X}$, such that

1. $x^* \in \mathcal{D}$;
2. F is α-convex and β-smooth in region $\mathcal{D}$;
3. $|\mathcal{D}|/|\mathcal{X}| \geq c/d$ for some $c > 0$.

A function F is α-convex in region $\mathcal{D}$, if $F(y) \geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2, x, y \in \mathcal{D}$. Furthermore, it is β-smooth, if $|F(y) - F(x) - \langle \nabla F(x), y - x \rangle| \leq \frac{\beta}{2}\|x - y\|^2, x, y \in \mathcal{D}$.

Theorem 1. For any $\alpha\beta$-Sampling-Easy function F that satisfies $F(x^*) \leq F(x) - \epsilon_0$, for all $x \notin \mathcal{D}$, by running Algorithm 3, on average it requires at most

$$O\left(\log\left(\frac{D_m \beta}{\min\{\epsilon, \epsilon_0\}}\right)\frac{d^2\beta}{c\alpha}\right)$$

iterations to find an ϵ-optimal solution for any $\epsilon > 0$, with c and d the same in Definition 1. Here $D_m = \max_{x \in \mathcal{D}} \|x - x^*\|_2^2$. The proof is provided in Appendix B.

Theorem 1 suggests that despite the function is non-convex or non-smooth, the convergence can be guaranteed as long as there is a sufficiently large convex and smooth area around the global optima.

B CONVERGENCE FOR ZEROTH-ORDER OPTIMIZATION

Algorithm 3 One-step Zeroth-Order Optimization with Consistent Iteration

Require
Objective function Q, domain $\mathcal{A}$, current point a_0, number of local samples n_1, number of global samples n_2, local scale $\eta > 0$ and step size h, number of steps m.
for $t = 1, \dots n_2$ **do**
  Globally sampling
  Sample a point uniformly in the entire space by

$$a_{t0} \sim \mathcal{U}_{\mathcal{A}}$$

  where $\mathcal{U}_{\mathcal{A}}$ is the uniform distribution over $\mathcal{A}$.
  for $i = 1, \dots, m$ **do**
    Locally sampling
    Sample n_1 points around $a_{t,i-1}$ by

$$\tilde{a}_j = a_{t,i-1} + \mu e_j \text{ for } e_j \sim \mathcal{N}(0, I_d), j = 1, \dots n_1,$$

    where $\mathcal{N}(0, I_d)$ is the standard normal distribution centered at 0.
    Update
    Set $a_{t,i} = a_{t,i-1} + h(\arg\max_{a \in \{\tilde{a}_j\}} Q(a) - a_{t,i-1})$
  end for
end for
return $\max_{a \in \{a_t m\}_{t=1}^{n_2}} Q(a)$.

Proof Sketch. As shown in Nesterov & Spokoiny (2017), under the same condition in Definition 1, given $x_0 \dots, x_N \in \mathcal{D}$, we have

$$F(x_N) - F(x^*) \leq \frac{\beta}{2}(1 - \frac{\alpha}{8(d+4)\beta})^N \|x_0 - x^*\|^2.$$

Thus, as long as there is a global sample lie in $\mathcal{D}$, it requires at most

$$N_\epsilon = \log(\frac{\beta \|x_0 - x^*\|^2}{2\epsilon}) \frac{8(d+4)\beta}{\alpha}$$

iterations to find an ϵ-optimal maxima.

The probability of sampling a point in $\mathcal{D}$ globally is at least $\frac{c}{d}$. On expectation, it requires $\frac{d}{c}$ global samples to start from a point in $\mathcal{D}$. Theorem 1 follows.

C ALGORITHM 4: ZOSPI WITH BOOTSTRAPPED Q NETWORKS

Algorithm 4 ZOSPI with UCB Exploration

Require

- The number of epochs M, the size of mini-batch N, momentum $\tau > 0$ and the number of Bootstrapped Q-networks K.
- Random initialized policy network π_{θ_1}, target policy network $\pi_{\theta'_1}$, $\theta'_1 \leftarrow \theta_1$.
- K random initialized Q networks, and corresponding target networks, parameterized by $w_{k,1}, w'_{k,1}$, $w'_{k,1} \leftarrow w_{k,1}$ for $k = 1, \ldots, K$.

for iteration $= 1, 2, ...$ **do**
  for t $= 1, 2, ..., T$ **do**
    # Interaction
    Run policy $\pi_{\theta'_t}$, and collect transition tuples $(s_t, a_t, s'_t, r_t, m_t)$.
    for epoch $j = 1, 2, ..., M$ **do**
      Sample a mini-batch of transition tuples $\mathcal{D}_j = \{(s, a, s', r, m)_i\}_{i=1}^N$.
      # Update Q
      for $k = 1, 2, ..., K$ **do**
        Calculate the k-th target Q value $y_{ki} = r_i + \max_l Q_{w'_{k,t}}(s'_i, a'_l)$, where $a'_l \sim \mathcal{U}_\mathcal{A}$.
        Update $w_{k,t}$ with loss $\sum_{i=1}^N m_{ik}(y_{ki} - Q_{w'_{k,t}}(s_i, a_i))^2$.
      end for
      # Update π
      Calculate the predicted action $a_0 = \pi_{\theta'_t}(s_i)$
      Sample actions $a_l \sim \mathcal{U}_\mathcal{A}$
      Select $a^+ \in \{a_l\} \cup \{a_0\}$ as the action with maximal $Q^+(s_t, a)$ defined in (4).
      Update policy network with Eq.(2).
    end for
    $\theta'_{t+1} \leftarrow \tau\theta_t + (1-\tau)\theta'_t$.
    $w'_{k,t+1} \leftarrow \tau w_{k,t} + (1-\tau)w'_{k,t}$.
    $w_{k,t+1} \leftarrow w_{k,t}$; $\theta_{t+1} \leftarrow \theta_t$.
  end for
end for