Q-Pensieve: Boosting Sample Efficiency of Multi-Objective RL Through Memory Sharing of Q-Snapshots

Wei Hung[1,2]*, Bo-Kai Huang[1]*, Ping-Chun Hsieh[1], and Xi Liu[3]

[1] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2] Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3] Applied Machine Learning, Meta AI, Menlo Park, CA, USA

* Equal Contribution

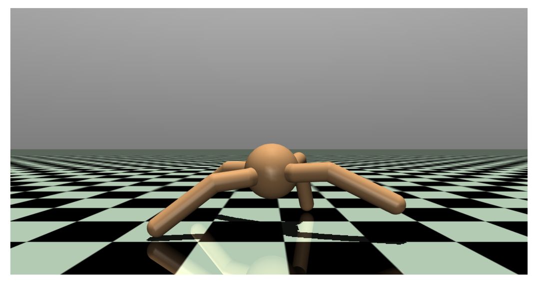
Introduction

- We identify the critical sample inefficiency issue in the existing MORL algorithms for continuous control
- We propose ***Q***-Pensieve, which is a policy improvement scheme for enhancing the data sharing capability across policies
- We substantiate the concept of ***Q***-Pensieve policy iteration by proposing the technique of ***Q*** replay buffer and arrive at a practical actor-critic type implementation
- We demonstrate that the proposed ***Q***-Pensieve can indeed achieve significantly better empirical sample

Multi-Objective Reinforcement Learning (MORL)

- Many real-world continuous control problems are also multi-objective tasks by nature, e.g., Control the robot to run fast while consuming as little energy as possible
- The multi-objective RL problems have been extensively studied from two major perspectives
  - Explicit search methods update a policy or a set of policies by explicitly searching for the Pareto front of the reward space (Xu et al., 2020; Kyriakis et al., 2022)
    - It is typically difficult to maintain a sufficiently diverse set of optimal policies for different preferences within a reasonable number of training samples
  - Implicit search methods improve policies for multiple preferences through implicit search (Abels et al., 2019; Yang et al., 2019)
    - Data sharing among policies of different preferences is not guaranteed to achieve policy improvement for all preferences

⇒ We want to propose an efficient way to solve multi-objective tasks.



Multi-Objective Soft Policy Iteration (MOSPI)

- With Soft Policy Iteration (SPI), we get a robust and good exploration method that can handle continuous actions properly
- Given a preference λ, we generalize the bellman operator to the multi-objective settings
- Multi-Objective Soft Policy evaluation:

$$(\mathcal{T}^{\pi}_{\mathrm{MO}}\boldsymbol{Q})(\boldsymbol{s},\boldsymbol{a};\lambda) = \boldsymbol{r}(\boldsymbol{s},\boldsymbol{a}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{P}(\cdot|\boldsymbol{s},\boldsymbol{a}),\boldsymbol{a}'\sim\pi(\cdot|\boldsymbol{s}';\lambda)}[\boldsymbol{Q}(\boldsymbol{s}',\boldsymbol{a}';\lambda) - \alpha\log\pi(\boldsymbol{a}'\mid\boldsymbol{s}';\lambda)\mathbf{1}_d]$$

- Multi-Objective Soft Policy improvement:

$$\pi_{k+1}(\cdot,\cdot;\lambda) = \arg\min_{\pi'\in\Pi} D_{KL}\Big(\pi'(\cdot\mid\boldsymbol{s})\Big\|\frac{\exp(\frac{1}{\alpha}\lambda^{\top}\boldsymbol{Q}^{\pi_k}(\boldsymbol{s},\cdot;\lambda))}{Z^{\pi_k}_{\lambda}(\boldsymbol{s})}\Big)$$

Envelope Q-Learning

- Align one preference with optimal rewards that may have been explored under other preferences (Yang et al., 2019)
- Optimality filter for multi-objective $\boldsymbol{Q}$:

$$(\mathcal{H}\boldsymbol{Q})(\boldsymbol{s};\lambda) = \arg_{\boldsymbol{Q}}\sup_{\boldsymbol{a}\in\mathcal{A},\lambda'\in\Lambda}\lambda^{\top}\boldsymbol{Q}(\boldsymbol{s},\boldsymbol{a};\lambda')$$

Q-Pensieve

- $\mathcal{Q}$ denotes all possible $\boldsymbol{Q}$, then we can achieve policy improvement according to

$$\max_{\boldsymbol{Q}(\boldsymbol{s})\in\mathcal{Q}}\lambda^{\top}\boldsymbol{Q}$$

- Intuition: Enforce knowledge sharing at the policy-level and enhance the sample used in learning a variety of Pareto optimal policies
- Implementation of ***Q***-Pensieve:
  Since we can directly use the ***Q*** functions of policies in the past without additional training, we store the learned ***Q*** network in the past iterations as candidates for forming the committee
- ***Q*** Replay Buffer:
  The policy update of ***Q***-Pensieve would involve both the current ***Q***-function and the ***Q***-snapshots from the past iterations. We introduce ***Q*** replay buffer, which could store multiple ***Q***-networks in a predetermined manner

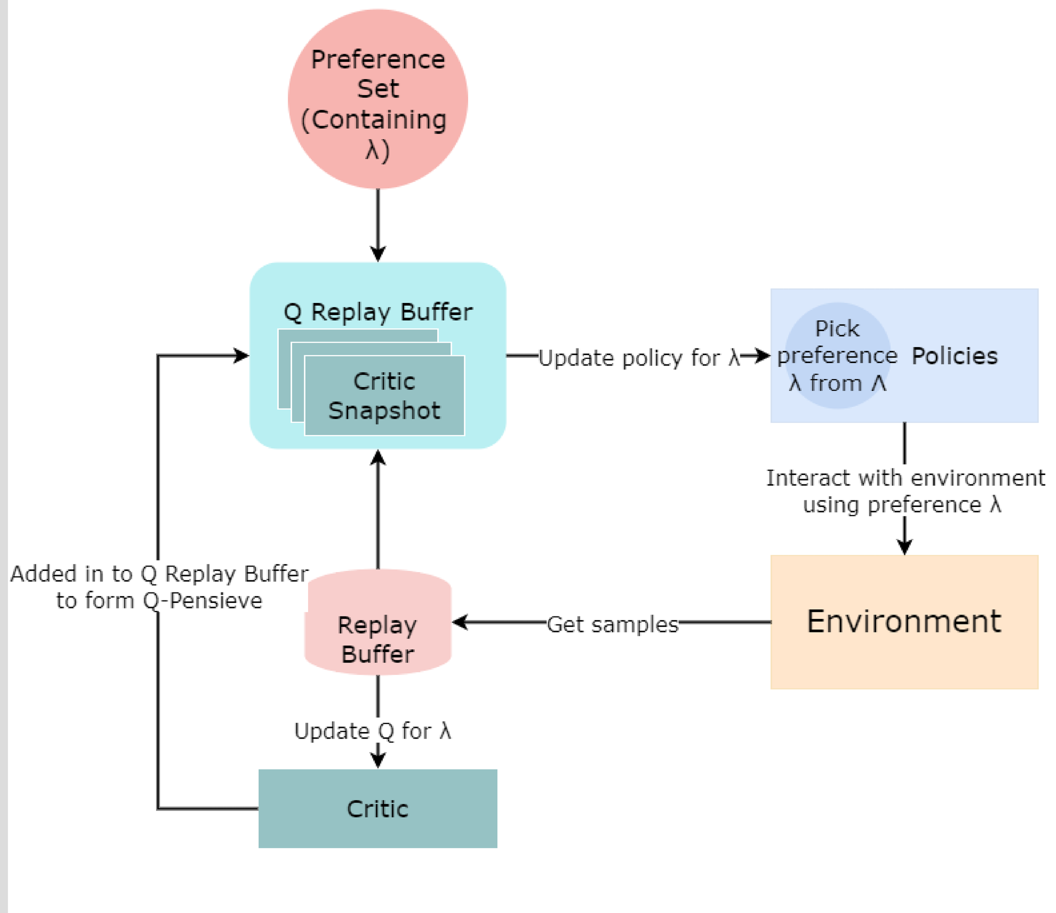
Q-Pensieve Soft Policy Iteration

- ***Q***-Pensieve Policy Improvement: In the policy improvement step of the k-th iteration, for each specific λ, we update the policy as

$$\pi_{k+1} = \arg\min_{\pi'\in\Pi} D_{KL}\Big(\pi'(\cdot\mid\boldsymbol{s};\lambda)\Big\|\frac{\exp\sup_{\lambda'\in W_k(\lambda),\boldsymbol{Q}'\in\mathcal{Q}_k}(\frac{1}{\alpha}\lambda^{\top}\boldsymbol{Q}'^{\pi_k}(\boldsymbol{s},\cdot;\lambda'))}{Z^{\pi_k}(\boldsymbol{s})}\Big)$$

- Convergence of ***Q***-Pensieve Soft Policy Iteration:
  Repeated application of soft policy evaluation and policy improvement to any $\pi\in\Pi$ converges to a policy π^* such that $\lambda^{\top}\boldsymbol{Q}^{\pi^*}(\boldsymbol{s},\boldsymbol{a};\lambda)\geq\lambda^{\top}\boldsymbol{Q}^{\pi}(\boldsymbol{s},\boldsymbol{a};\lambda)$ for all $(\boldsymbol{s},\boldsymbol{a})\in\mathcal{S}\times\mathcal{A}$ and all $\lambda\in\Lambda$

Algorithm



Q-Pensieve algorithm: In each iteration, do the following:
(1) Pick a preference λ, interact with the environment, and add samples to the replay buffer
(2) Update the critic network by minimizing the critic loss $\mathcal{L}_{\boldsymbol{Q}}$
(3) Build up a preference set $\boldsymbol{W_k}(\lambda)$ with λ, sample N preferences and add them to $\boldsymbol{W_k}(\lambda)$
(4) Update the ***Q*** replay buffer $\mathcal{Q}_k$ with current critic network
(5) Update the policy network by minimizing the policy loss $\mathcal{L}_{\pi}$

Figure: The architecture of Q-Pensieve

$$\mathcal{L}_{\boldsymbol{Q}}(\phi;\lambda) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\mu}\Big[\lambda^{\top}\big(\boldsymbol{Q}_{\phi}(\boldsymbol{s},\boldsymbol{a};\lambda) - \big(\boldsymbol{r}(\boldsymbol{s},\boldsymbol{a}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim\mathcal{P}(\cdot|\boldsymbol{s},\boldsymbol{a})}\big[V_{\bar{\phi}}(\boldsymbol{s}')\big]\big)\big)^2\Big]$$

$$\mathcal{L}_{\pi}(\theta;\lambda) = \mathbb{E}_{\boldsymbol{s}\sim\mu}\Big[\mathbb{E}_{\boldsymbol{a}\sim\pi_{\theta}}\Big[\sup_{\lambda'\in W_k(\lambda),\boldsymbol{Q}\in\mathcal{Q}_k}\{\alpha\log(\pi_{\theta}(\boldsymbol{a}\mid\boldsymbol{s};\lambda)) - \lambda^{\top}\boldsymbol{Q}_{\phi}(\boldsymbol{s},\boldsymbol{a};\lambda')\Big]\Big]$$

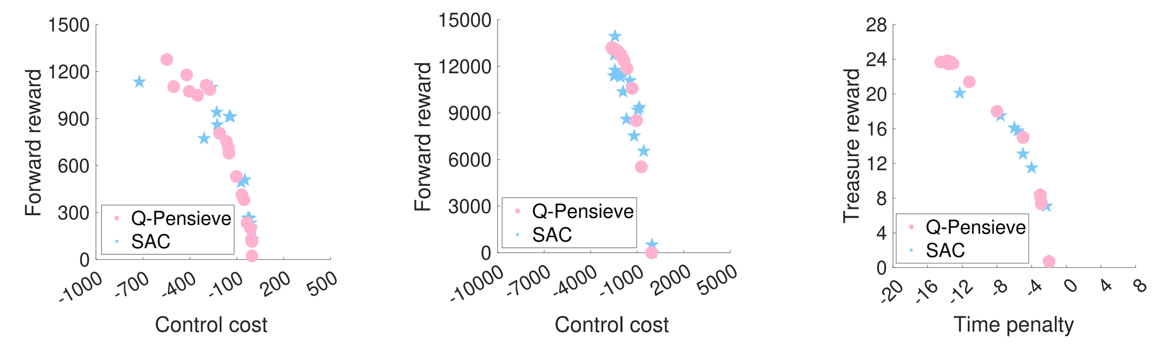
Experimental Results - Comparison with Benchmarks

- **Performance Metrics**
  - HyperVolume (HV): Compute the region dominated by the solution set w.r.t. a given reference point $\boldsymbol{r_0}$
  - Utility (UT): Evaluate the performance under linear scalarization

Environments	Metrics	PFA (Parisi et al., 2014)	PGMORL (Xu et al., 2020)	CN-DER (Abels et al., 2019)	***Q*-Pensieve**
DST2d	HV($\times 10^2$)	7.43	8.10	5.36	**10.21**
	UT($\times 10^0$)	-9.27	4.90	-5.10	**7.31**
HalfCheetah2d	HV($\times 10^7$)	0.73	0.53	2.08	**3.82**
	UT($\times 10^3$)	0.31	-0.28	5.09	**5.61**
Ant3d	HV($\times 10^8$)	-	0.41	13.00	**21.87**
	UT($\times 10^3$)	-	0.18	0.49	**1.14**
Hopper5d	HV($\times 10^{13}$)	-	0.63	3.42	**7.24**
	UT($\times 10^2$)	-	1.48	1.76	**3.37**

Experimental Results - Sample Efficiency of Q-Pensieve

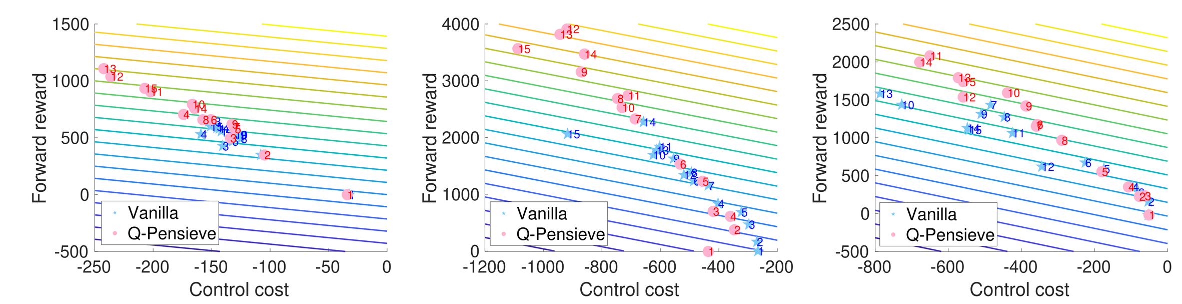
We train 19 single-objective SAC models, each for a unique preference, and train ***Q***-Pensieve for all preferences. The following figures show the return vectors attained by ***Q***-Pensieve and the collection of single-objective SAC models under 19 preferences.



(a) Hopper2d (b) HalfCheetah2d (c) DST2d

An Ablation Study on Q Replay Buffer

To verify the effectiveness of the technique of ***Q*** replay buffer, we compare the performance of ***Q***-Pensieve with buffer size equal to 4 and that without using ***Q*** replay buffer (termed "Vanilla"). The following figures show the return vectors attained under preference $\lambda = [\mathbf{0.5}, \mathbf{0.5}]$ at $\mathbf{100}\cdot\boldsymbol{x}$ thousand training steps.



(a) Hopper2d (b) Ant2d (c) Walker2d

Conclusion

- We propose ***Q***-Pensieve to boost the sample efficiency of MORL problems
- We present ***Q***-Pensieve soft policy iteration in the tabular setting and show that it preserves the global convergence property
- Our theoretical and experimental results demonstrate that the proposed learning algorithm is indeed a promising approach for MORL problems

hwei1048576.cs08@nycu.edu.tw