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], Xi Liu[3]*

ICLR 2023

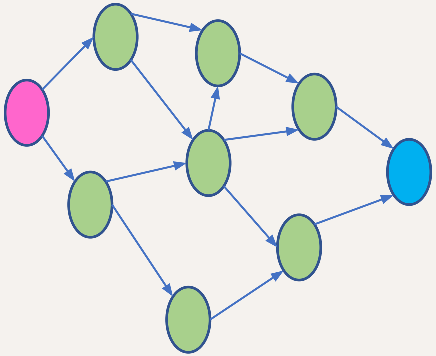
1. Department of Computer Science, National Yang Ming Chiao Tung University
2.Research Center for Information Technology Innovation, Academia Sinica
3.Applied Machine Learning, Meta AI

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 practical implementation

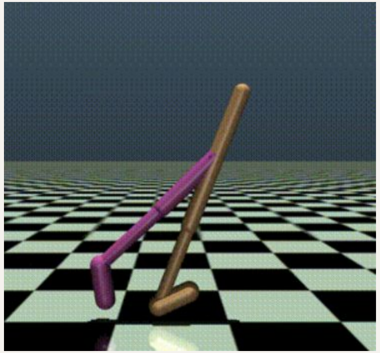
Multi-Objective Applications in Real World

Communication networks



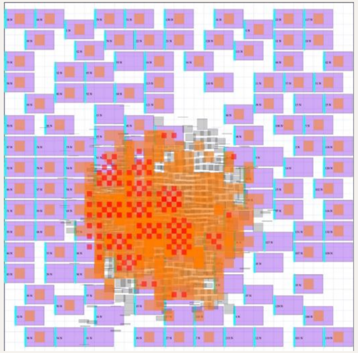
- Throughput and latency

Robotics



- Speed, survival bonus and control cost

Chip Design



- Wirelength, routing congestion, and density

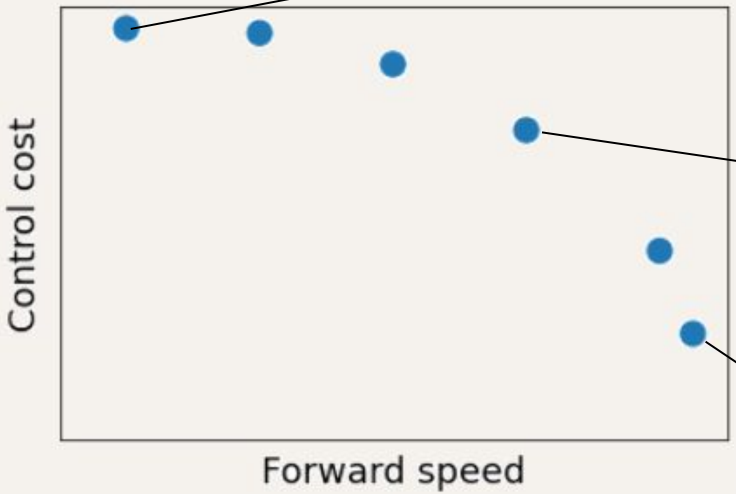
(Credit by Mirhoseini et al.)

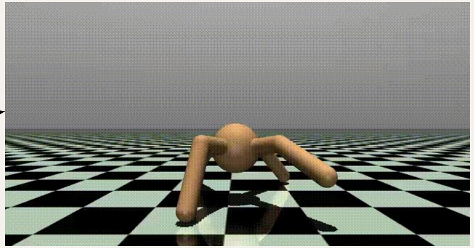
1. https://www.nature.com/articles/s41586-021-03544-w.

MORL Learns Diverse Behaviors

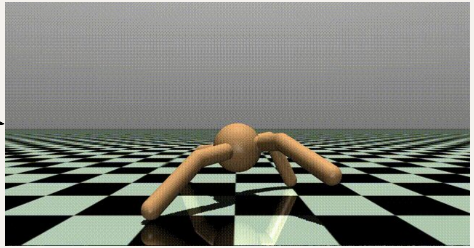
Forward speed: running speed

Control cost: -energy consumption

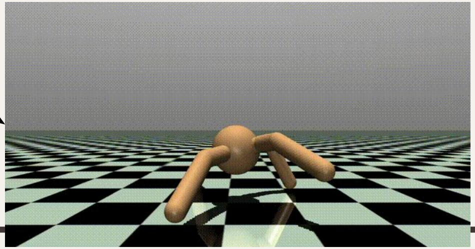




λ=[0.1,0.9]
Balance with minimal contro
cost

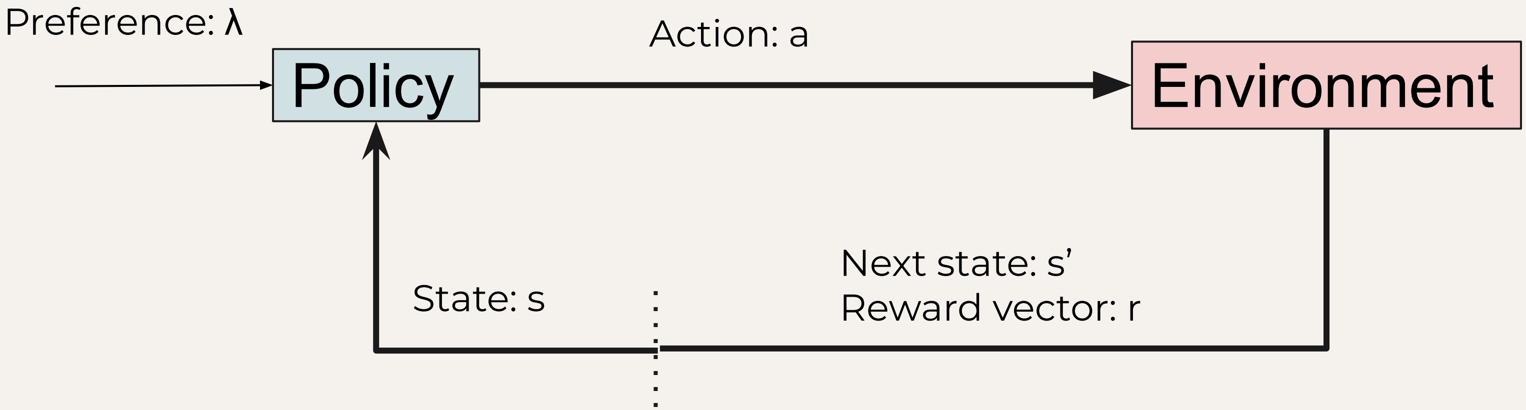


λ=[0.5,0.5]
Walk gracefully



λ=[0.9,0.1]
Sprint at all cost!!

MORL Formulation



$$Q_{\pi_\theta}(s,a) := \mathbf{E}[\Sigma_{t=0}^{\infty}\gamma^t \mathbf{r}_t | s_0 = s, a_0 = a; \pi_\theta]$$ (Q-functions in MORL)

$$J(\pi_\theta) := \mathbf{E}_{s_0, a_0 \sim \pi_\theta}[Q_{\pi_\theta}(s_0, a_0)]$$ (Vector-Valued Total Return)

Goal: Given all λ, learn π_θ that maximizes utility $\lambda^\intercal J(\pi_\theta)$

Convex Coverage Set (CCS)

: Non-CCS

It is a total return vector of some π

$$J(\pi) = \Sigma_t \gamma^t \mathbf{r}(s,a)$$

: CCS

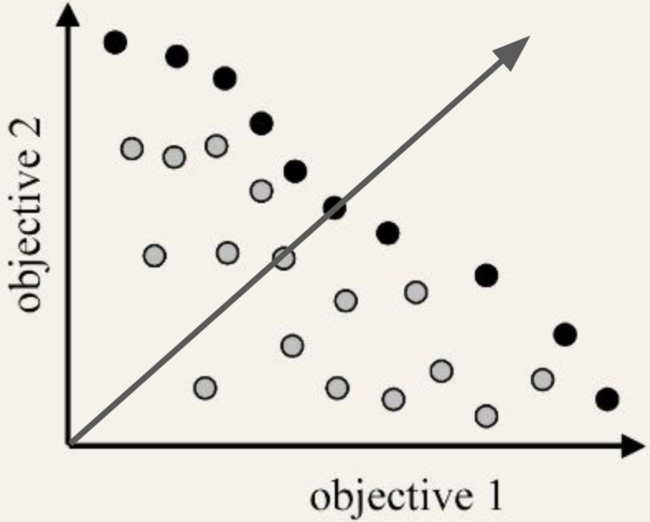
It is an optimal return vector for some λ

$$\lambda^\intercal J(\pi^*) \geq \lambda^\intercal J(\pi), \forall \pi$$

Different Aspects:

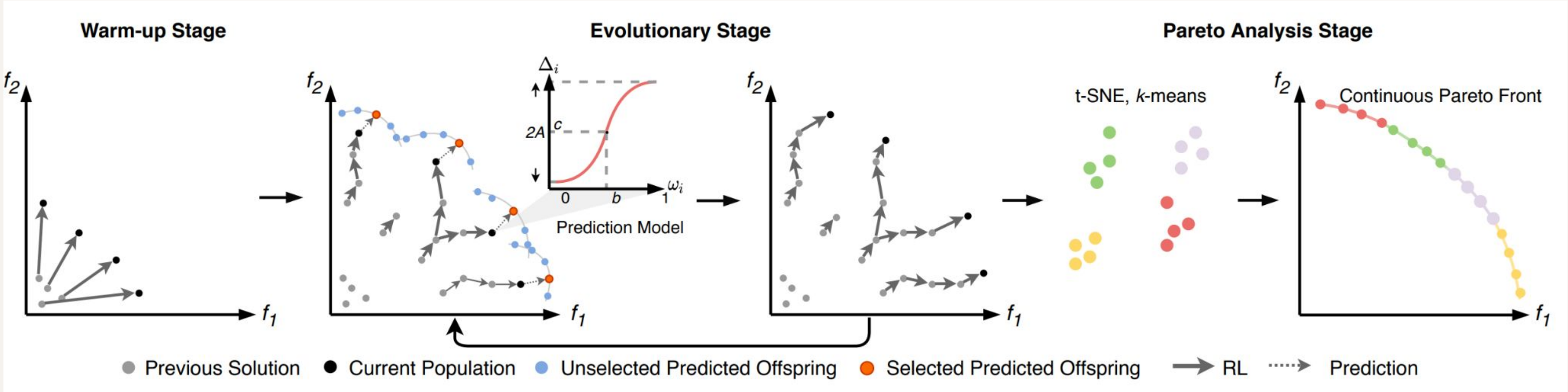
- Maximize hypervolume (Area covered by reward vector)
- Maximize $\lambda^\intercal J(\pi), \forall \lambda$

same preference
(λ=[0.6, 0.4])



Existing Solutions to MORL

- Explicit search - PGMORL
  - Evolutionary search for CCS
  - Issue: Sample inefficiency!

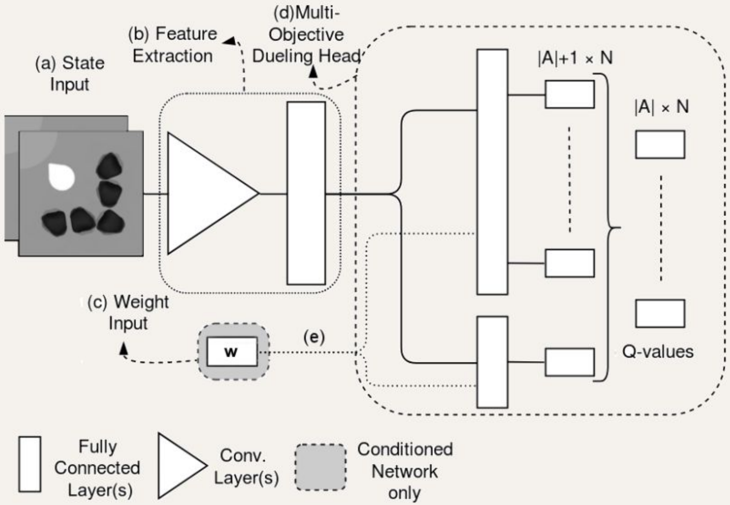


[Xu et al.,2020]

1. Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control, ICML 2020

Existing Solutions to MORL

- Implicit search - Conditioned Networks (CN)
  - Preference-dependent MO-DQN
  - Issue: No policy improvement guarantee!



[Abels et al.,2019]

1. Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic Weights in Multi-Objective Deep Reinforcement Learning, ICML 2019

Envelope Q-Learning

- Align one preference with optimal rewards that may have been explored under other preferences
- Bellman backup operator

$$\mathcal{T}_\pi Q(s, a; \lambda) = \mathbf{r}(s, a) + \gamma \mathbf{E}_{(s', a') \sim (\mathcal{P}, \pi)} Q(s', a'; \lambda)$$

- Optimality filter for multi-objective Q

$$(\mathcal{H}Q)(s; \lambda) = \arg_Q \sup_{\underline{a \in \mathcal{A}, \lambda' \in \Lambda}} \lambda^\intercal Q(s, a; \lambda')$$

1. Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation, NeurIPS 2019

Our Idea: Memory Sharing of Snapshots (Called Q-Pensieve)

$Q_{\pi_3}(s, a, \lambda)$

$Q_{\pi_2}(s, a, \lambda)$

$Q_{\pi_1}(s, a, \lambda)$

Training Iterations

- Policy-level knowledge sharing: Snapshots can boost the learning in future iterations
- Each Q-network $Q_{\pi_k}(s, a, \lambda)$ can be good for some preference vector λ

Pensieve: A magical device used to review and store memories

Figure credied by https://screenrant.com/harry-potter-pensieve-memories-movies-missing/.

Q-Pensieve Policy Improvement Update

MO Soft Policy improvement

$$\pi_{k+1}(\cdot|\cdot;\lambda) = \arg\min_{\pi' \in \tilde{\Pi}} \mathrm{D}_{\mathrm{KL}}\left(\pi'\left(\cdot \mid s;\lambda\right) \middle\| \frac{\exp\left(\sup_{\lambda' \in W_k(\lambda), \mathbf{Q}' \in \mathcal{Q}_{\mathbf{k}}} \left(\frac{1}{\alpha}\lambda^\top \mathbf{Q}'\left(\cdot,\cdot;\lambda'\right)\right)\right)}{Z_{\mathcal{Q}_k}(s)}\right)$$

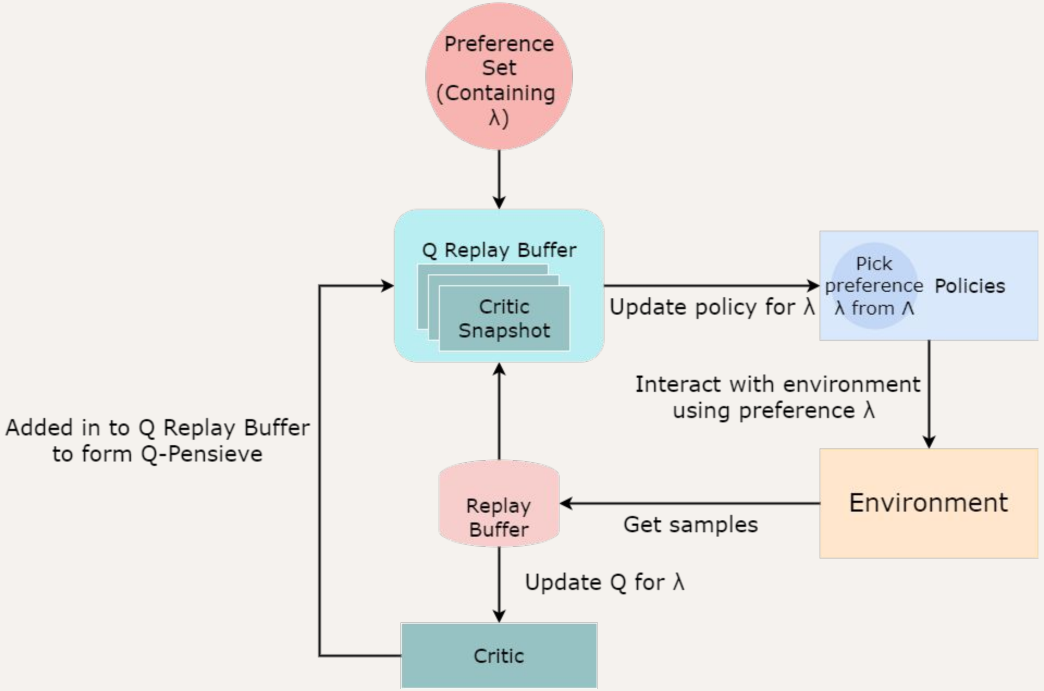
W□(λ) is a subset of Λ containing λ

$\mathcal{Q}$□(λ) is a set of Q-snapshots containing $Q_{\pi_k}(s,a,\lambda)$

Theorem (Convergence of Q-Pensieve)

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

Implementation of Q-Pensieve

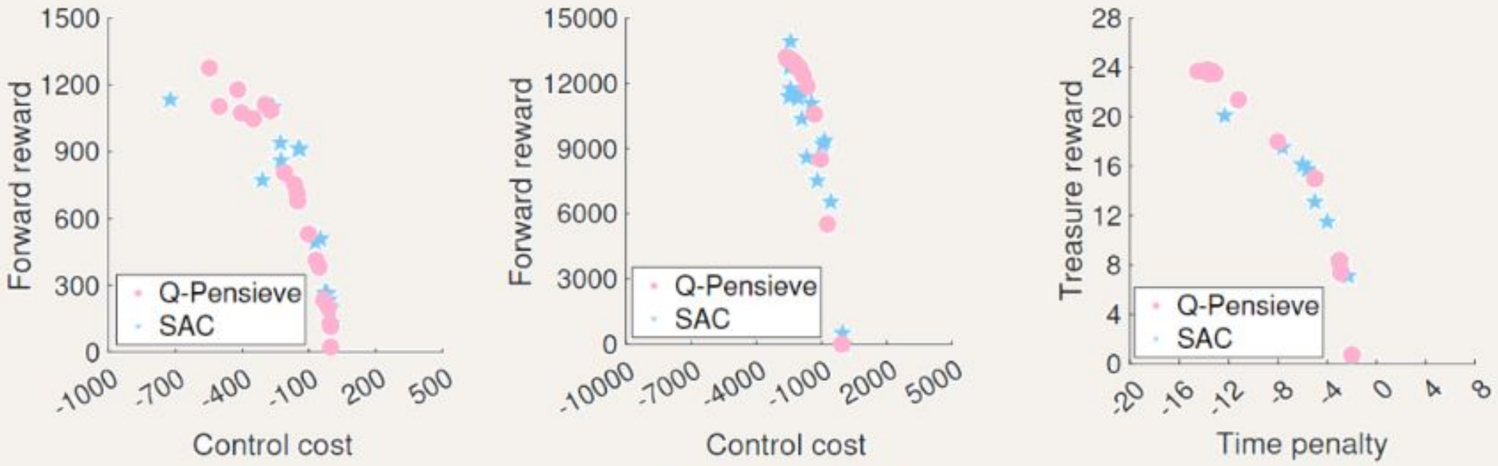


Experimental Results - Comparison with Baselines

Environments	Metrics	PFA (1.5M steps)	PFA (1.5×βM steps)	PGMORL (1.5M steps)	PGMORL (1.5×βM steps)	CN-DER (1.5M steps)	Q-Pensieve (1.5M steps)
DST2d	HV($\times 10^2$)	7.43±3.68	8.67±1.49	8.10±1.57	8.13±1.61	5.36±4.71	**10.21±1.40**
	UT	-9.27±6.03	-6.86±6.06	4.90±0.44	5.02±0.35	-5.10±15.73	**7.31±0.91**
	ED	0.13±0.11	0.10±0.08	0.25±0.18	0.28±0.18	0.21 ±0.17	**0.54±0.11**
HalfCheetah2d	HV($\times 10^7$)	0.73±0.19	1.31±0.26	0.53±0.17	0.28±0.29	2.08±0.54	**3.82±0.27**
	UT($\times 10^3$)	0.31±0.20	1.02±0.40	-0.28±0.94	0.09±0.17	5.09±3.57	**5.61±0.31**
	ED	0.08±0.10	0.10±0.06	0.01±0.00	0.11±0.05	0.02 ±0.01	**0.54±0.08**
Hopper2d	HV($\times 10^6$)	0.49±0.46	1.01±0.62	0.63±0.48	1.31±0.48	0.56±0.16	**1.33±0.20**
	UT($\times 10^2$)	2.89±1.93	3.50±1.85	1.94±2.46	3.70±1.78	1.42±1.00	**4.08±1.10**
	ED	0.31±0.17	0.41±0.10	0.31±0.25	0.31±0.11	0.04 ±0.03	**0.43±0.09**
Ant2d	HV($\times 10^6$)	0.17±0.05	0.77±0.53	0.14±0.03	0.13±0.04	5.03±3.60	**10.01±1.86**
	UT($\times 10^2$)	-0.06±0.01	0.14±0.14	-0.21±0.15	-0.18±0.38	3.68±2.34	**14.04±3.03**
	ED	0.22±0.03	0.22±0.02	0.21±0.02	0.21±0.03	0.21±0.08	**0.60±0.07**
Walker2d	HV($\times 10^6$)	0.52 ±0.20	1.05±0.44	0.83±0.42	**1.28±0.66**	0.42±0.09	1.12±0.36
	UT($\times 10^2$)	0.23±0.13	0.95±0.55	0.38±0.24	1.20±0.67	3.17±0.53	**6.37±1.42**
	ED	0.32±0.06	0.37±0.09	0.30±0.10	0.34±0.12	0.21 ±0.11	**0.48±0.10**

Experimental Results - Sample Efficiency of Q-Pensieve

- Select 19 fixed preferences and learn 19 separate SAC models
- We achieve the same performance with only 1/19 of samples used by SAC



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

Summary

- 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