南京航空航天大學
Nanjing University of Aeronautics and Astronautics

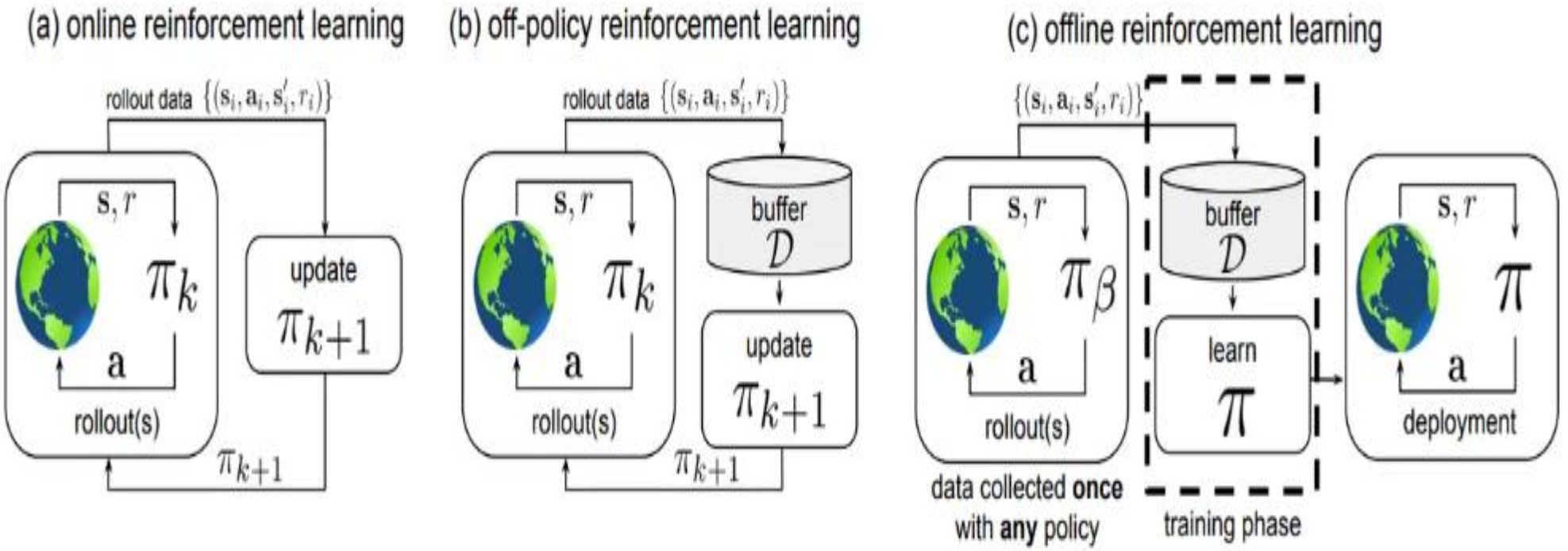
模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

Diffusion-based Reinforcement Learning via Q-weighted Variational Policy Optimization

NIPS | 2024

Background

- **Reinfocement learning**



Background

- **Gaussian Policies for reinforcement learning**
  - Used in continuous reinforcement learning for action generation.

$$\pi_\theta(\mathbf{a} \mid \mathbf{s}) = \mathcal{N}(\mu_\theta(\mathbf{s}), \Sigma_\theta(\mathbf{s}))$$

  - **advantages:**
    - Differentiable
    - Adapting to reinforcement learning(std for exploration)

  - **Disadvantages:**
    - One peak
    - Limited distribution

Background

- **Diffusion Model**
  - Forward process: add noise to action to generate a standard Gaussian distribution

$$q(a_t \mid a_{t-1}) = \mathcal{N}\left(a_t; \sqrt{1-\beta_t}a_{t-1}, \beta_t I\right)$$

$$a_t = \sqrt{\bar{\alpha}_t}a_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

  - reverse process: A neural network fθ is trained to denoise the action

$$p_\theta(a_{t-1} \mid a_t) = \mathcal{N}\left(a_{t-1}; \mu_\theta(a_t, t), \Sigma_\theta(a_t, t)\right)$$

Background

- **Diffusion Model & Bayesian Estimation**

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

$$p(a_{t-1}|a_t) = \frac{p(a_t|a_{t-1})p(a_{t-1})}{p(a_t)}$$

  - **What we cant know:**
    - Distribution of a_t
    - Distribution of a_t-1
  - **What we can know:**
    - Groundtruth of t timestep noise.
    - Noise distribution(sample)

- **Diffusion Model optimization objective**
  (VLO)

$$\log p(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\sum_{t=1}^{T} -D_{KL}(q(x_t|x_{t-1})||p_\theta(x_t|x_{t-1})) \right]$$

- **Diffusion Model for RL**

$$a_t = \sqrt{\bar{\alpha}_t} a_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

$$a_0 \approx \frac{a_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}_\theta(a_t, t)}{\sqrt{\bar{\alpha}_t}}$$

Background

- **Process of Diffusion Model Training**
  - **Purpose**
    - Train a neural network ϵθ(at,s,t) to predict noise, and to further denoise.
  - Process
    - Sample(s,a) & time step(T)(uniform)
    - Generate noise ϵ~ N(0, I)
    - Calculate at based on $a_t = \sqrt{\bar{\alpha}_t} a_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$
    - Predict ϵ’ using ϵθ(at,s,t)
- **Motivation**
  - **Diffusion model commonly used for offline RL**
  - **Diffusion cant use directly in Online RL(bad data)**
  - **a diffusion-based algorithm for online RL**

Contribution

The paper dedicated to find a solution to deploy diffusion into online RL. The contributions are:

- Q-weighted VLO Loss
  构造一个基于q值加权的损失函数，确保diffusion生成的动作匹配最优q函数的方向
- Diffusion Entropy Regularization
  引入熵正则化项，鼓励探索
- Efficient Behavior Policy
  提高样本利用率

Methodology

- **Q-weighted Variational Objective for Diffusion Policy**

Theorem 1. ***(Lower Bound of RL Policy Objective)*** *If* $Q(s,a) \geq 0$ *for any state-action pair* (s,a)*, the Q-weighted variational bound objective of diffusion policy*

$$\mathbb{E}_{s,a\sim p(s'|s,a),\pi_k(a|s)}\left[Q(s,a_0)\cdot \mathbb{E}_{a_{1:T}\sim q(a_{1:T}|s,a_0)}\left[\log \frac{\pi_\theta\left(a_{0:T}\mid s\right)}{q\left(a_{1:T}\mid s,a_0\right)}\right]\right]$$

is the tight lower bound of the objective of RL policy

$$\mathbb{E}_{s,a\sim p(s'|s,a),\pi_k(a|s)}\left[Q(s,a)\log(\pi_\theta(a|s))\right],$$

and the equality holds when the policy converges.

作者发现，对 VLO 目标**添加合适的权重**（即 Q 加权），那么在某些特定条件下，它可以成为**强化学习策略优化目标的一个紧下界（tight lower bound）**

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{s,a\sim\pi_k(a|s),\boldsymbol{\epsilon},t}\left[Q(s,a)\cdot\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}a+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},s,t\right)\right\|^2\right]. \tag{5}$$

Methodology

- **Equivalent Transformation for Q-weighted VLO Loss**

Issues:

- Negative Q-value
- High-quality Training Sample

Solutions:

$$L(\theta) \triangleq \mathbb{E}_{s,a\sim\pi_k(a|s),\epsilon,t}\left[\omega_{eq}(s,a)\cdot\left\|\epsilon-\epsilon_\theta\left(\sqrt{\bar{\alpha}_t}a+\sqrt{1-\bar{\alpha}_t}\epsilon,s,t\right)\right\|^2\right]$$

Q_cut weight:

$$p(a|s)=\begin{cases}\frac{\pi(a|s)Q(s,a)}{\int_{a'}\pi(a'|s)Q(s,a')da'}, & \text{if } \max_a Q(s,a)>0\\ \frac{1}{N}\sum_{i=1}^{N}\delta(x-a_i), \quad a_i\in\{a\mid Q(s,a)=\max_a Q(s,a),\pi(a|s)>0\}, & \text{if } \max_a Q(s,a)\le 0\end{cases}$$

Methodology

Q_adv weight:

$$\omega_{eq}(s,a) \triangleq \omega_{qadv}(s,a) = \begin{cases} A(s,a), & A(s,a) \geq 0 \\ 0, & A(s,a) < 0 \end{cases}$$

where $A(s,a) = Q(s,a) - V(s)$ is the advantage function.

- **Enhancing Policy Exploration via Diffusion Entropy Regularization**

Issues:

less diffusion steps, less calculation, less exploration(need a way to balance)

Solution:

entropy regularization

Methodology

Normal regularization doesn't fit diffusion

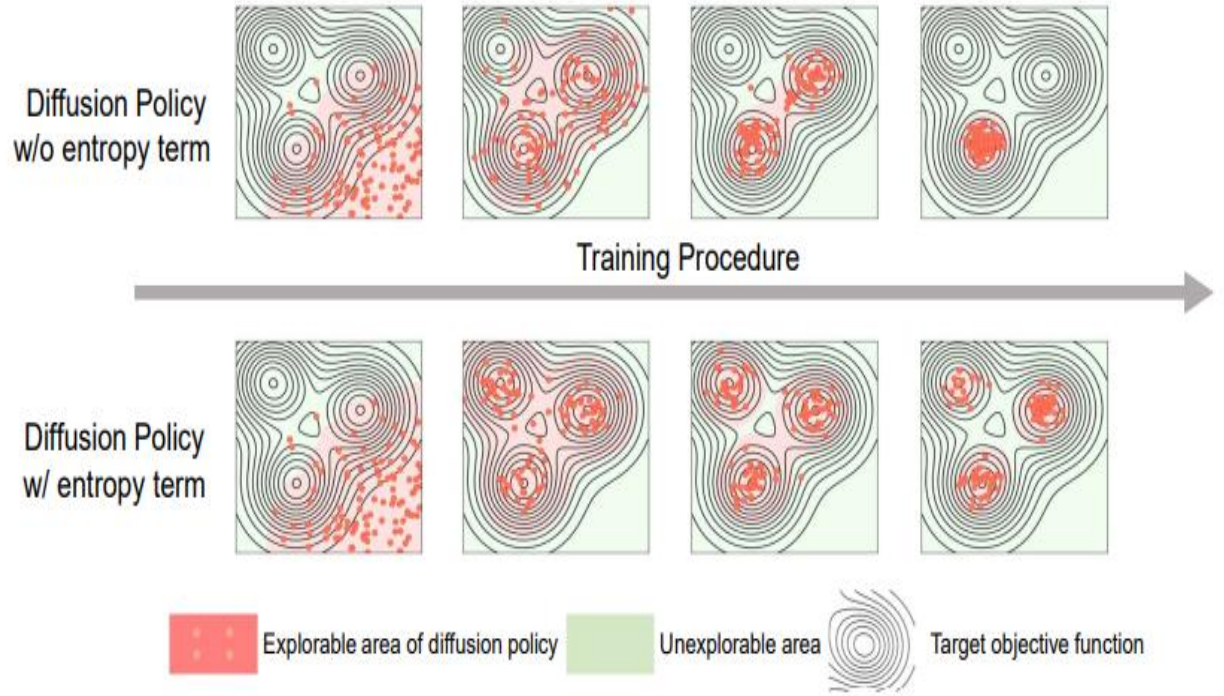
- Inaccessibility of Log-Likelihood
- Implicit policy representation
- Entropy is not straightforward

"the entropy of a diffusion model can be increased with training samples from the uniform distribution"

$$L_{\text{ent}}(\theta) \equiv \mathbb{E}_{s,a\sim U,\epsilon,t}\left[\omega_{\text{ent}}(s)\left(\left(\epsilon - \epsilon_\theta\sqrt{\bar{\alpha}_t}\right.\right.\right.$$

$$\left.\left.\left.+\sqrt{1-\bar{\alpha}_t}\epsilon, s, t\right)^2\right)\right]$$

where $\omega_{ent}(s) = \omega_{ent} \sum_{i=1}^{N} \frac{\omega_{eq}(s,a_i)}{N}$



igure 2: A toy example on continuous bandit to show the effect of diffusion entropy regularization rm via the changes of the explorable area for diffusion policy with the training procedure. The ontour lines indicate the reward function of continuous bandit, which is an arbitrarily selected function with 3 peaks.

Methodology

Normal regularization doesn't fit diffusion

- Inaccessibility of Log-Likelihood
- Implicit policy representation
- Entropy is not straightforward

"the entropy of a diffusion model can be increased with training samples from the uniform distribution"

$$L_{\text{ent}}(\theta) \equiv \mathbb{E}_{s,a\sim U,\epsilon,t}\left[\omega_{\text{ent}}(s)\left(\left(\epsilon - \epsilon_\theta\sqrt{\bar{\alpha}_t}\right.\right.\right.$$

$$\left.\left.\left.+\sqrt{1-\bar{\alpha}_t}\epsilon, s, t\right)^2\right)\right]$$

where $\omega_{ent}(s) = \omega_{ent}\sum_{i=1}^{N}\frac{\omega_{eq}(s,a_i)}{N}$

- **Reducing Diffusion Policy Variance via Action Selection**

$$\pi_\theta^K(a \mid s) \triangleq \underset{a\in\{a_1,\cdots,a_K\sim\pi_\theta(a|s)\}}{\operatorname{argmax}} Q(s,a).$$

1. 采样动作
2. 计算q值
3. 选q值高的K个动作

Methodology

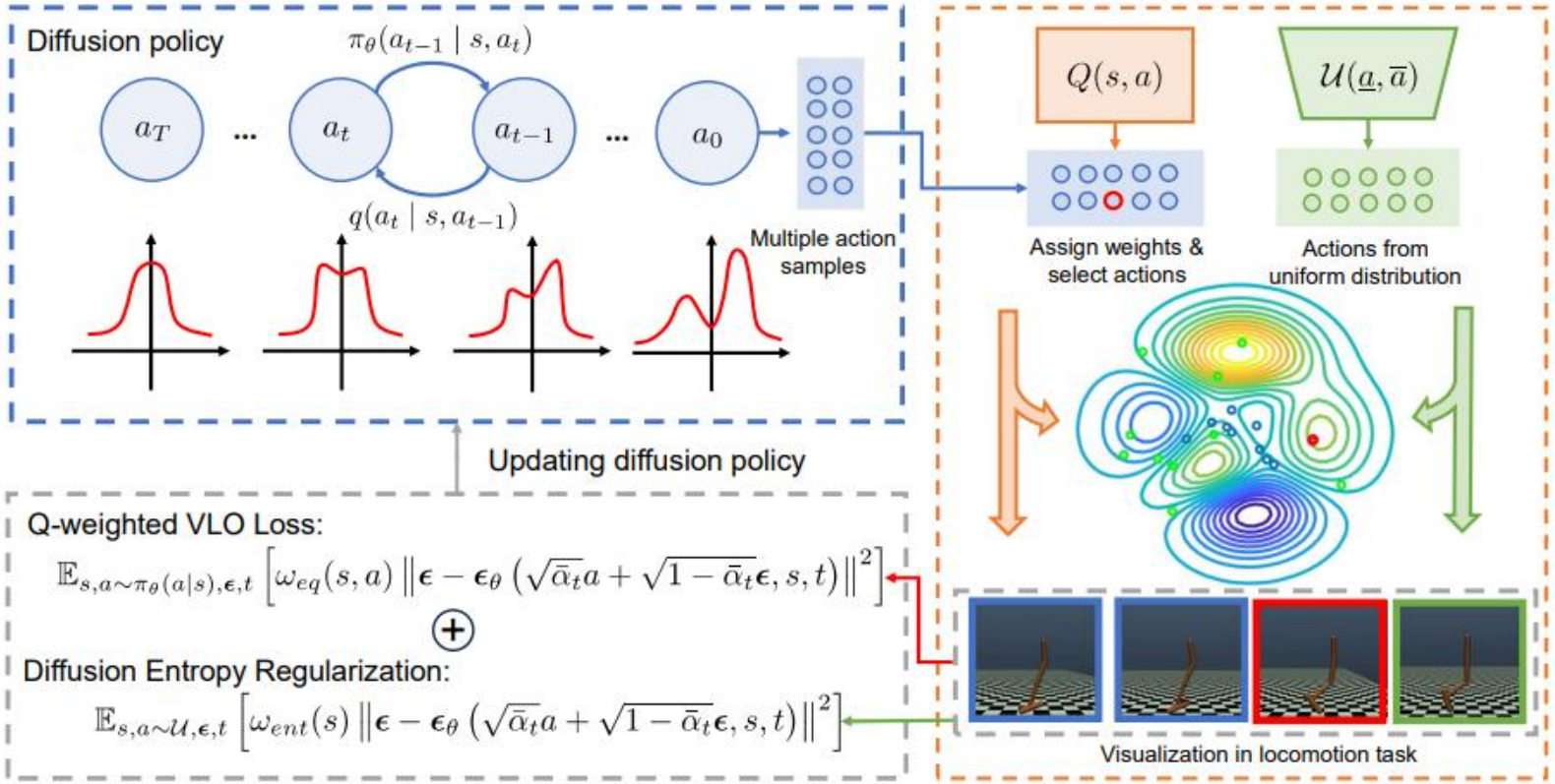


Figure 1: The training pipeline of QVPO. In each training epoch, QVPO first utilizes the diffusion policy to generate multiple action samples for every state. Then, these action samples will be selected and endowed with different weights according to the Q value given by the value network. Besides, action samples from uniform distribution are also created for the diffusion entropy regularization term. With these action samples and weights, we can finally optimize the diffusion policy via the combined objective of Q-weighted VLO loss and diffusion entropy regularization term.

- 用扩散模型生成多个候选动作（同一个s）。
- 利用 Q 值筛选并加权。
- 加入均匀分布的动作，通过熵正则化提升探索能力。
- 优化策略

Methodology

Algorithm 1 Q-weighted Variational Policy Optimization

Input: Diffusion policy $\pi_\theta(a \mid s)$, value network $Q_\omega(s, a)$, replay buffer $\mathcal{D}$, K_b-effiecient diffusion policy for behavior policy, K_t-effiecient diffusion policy for target policy, number of training samples N_d from diffusion policy, number of training samples N_e from uniform distribution $\mathcal{U}(\underline{a}, \overline{a})$.

for t **in** $1, 2, \cdots, T$ **do**
  Sample the action using the diffusion policy $\pi_\theta^{K_b}(a \mid s_t)$.
  Take the action a_t in the environment and store the returned transition in $\mathcal{D}$.
  Sample a mini-batch $\mathcal{B}$ of transitions in $\mathcal{D}$.
  Generate N_d samples from $\pi_\theta(a \mid s)$, and N_e samples from $\mathcal{U}(\underline{a}, \overline{a})$ for each state s in $\mathcal{B}$.
  Endow the N_d samples with weights (9).
  Select an action sample a_{max} with maximum weight among N_d samples for training.
  Endow the N_e samples with the weight $\omega_{ent}(s) = \omega_{ent} \cdot \omega_{eq}(s, a_{max})$.
  Update the parameters of the diffusion policy using the summation of (6) and (10).
  Construct TD target as $y_t = r_t + \gamma Q_\omega(s_{t+1}, \pi_\theta^{K_t}(a \mid s_{t+1}))$ for each (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$.
  Update the parameters of the value network using MSE loss.
end for

Experiments

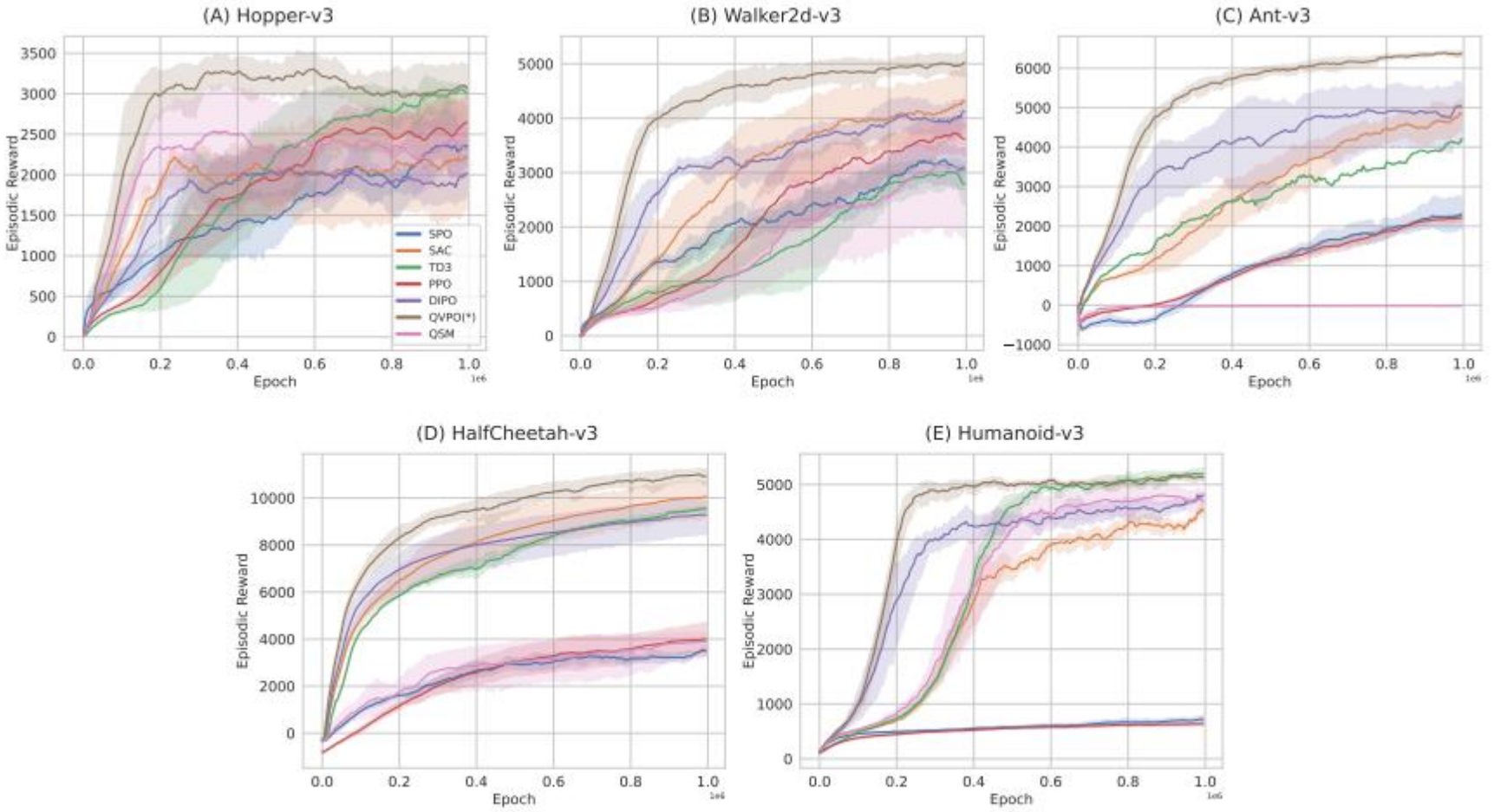


Figure 3: Learning Curves of different algorithms on 5 Mujoco locomotion benchmarks across 5 runs. The x-axis is the number of training epochs. The y-axis is the episodic reward. the plots smoothed with a window of 5000.

Experiments

Envrironments	PPO	SPO	TD3	SAC	DIPO	QSM	QVPO(*)
Hopper-v3	3154.3(426.2)	2212.8(988.4)	3267.5(8.5)	2996.6(111.9)	3295.4(7.0)	2154.7(998.2)	**3728.5(13.8)**
Walker2d-v3	3751.5(609.1)	3321.8(1328.9)	3513.9(40.7)	4888.1(80.0)	4681.7(25.7)	3613.4(1443.5)	**5191.8(60.2)**
Ant-v3	2781.9(74.1)	2100.2(302.4)	4583.8(69.5)	5030.9(1000.3)	5665.9(54.7)	N/A	**6425.1(67.6)**
HalfCheetah-v3	4773.5(53.4)	4008.2(246.8)	10388.6(80.4)	10616.9(72.8)	9590.5(67.5)	3888.2(632.6)	**11385.6(164.5)**
Humanoid-v3	713.7(85.9)	797.4(262.1)	**5353.5(53.7)**	5159.7(475.3)	4945.5(898.6)	4793.1(229.5)	**5306.6(14.5)**

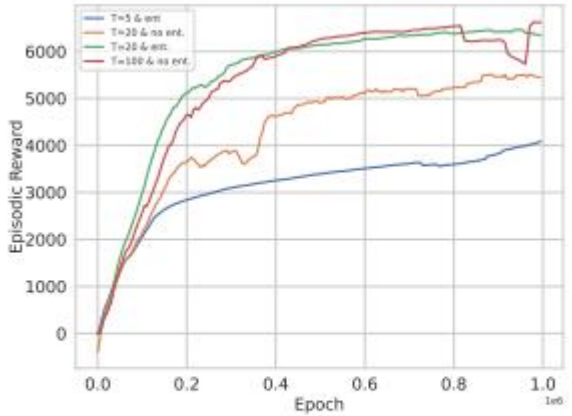


Figure 4: Comparison between QVPO with and without the diffusion entropy regularization.

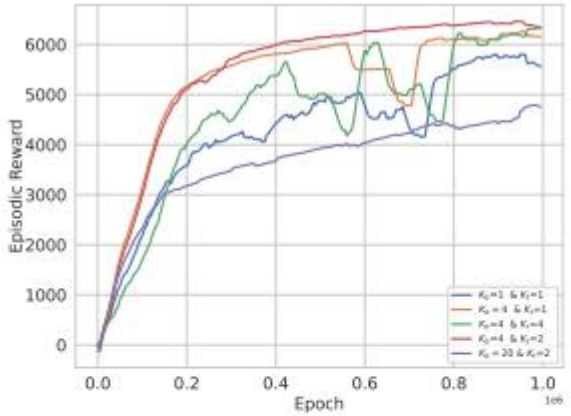


Figure 5: Comparison of QVPO with different action selection numbers for behavior policy K_b and for target policy K_t.