

A UPPER BOUND OF THE OPTIMIZATION OBJECTIVE

The details about how to derive the form of distillation is shown as follow:

$$\begin{aligned}
L_{\pi_o} &= \mathbb{E}_{s \sim \rho^{\pi_o}} [D_{KL}(\pi_{\theta_o}(\cdot|s) || \frac{\exp(\frac{1}{\alpha} Q_o^{\pi_o}(\cdot|s))}{Z(s)})] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [D_{KL}(\pi_{\theta_o}(\cdot|s) || \frac{\exp(\frac{1}{\alpha} \sum w_i Q_i^{\pi_o}(\cdot|s))}{Z(s)})] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [\int_a \pi_{\theta_o}(a|s) (\log \pi_{\theta_o}(a|s) - \frac{1}{\alpha} \sum w_i(s) Q_i^{\pi_o}(a|s))] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [\int_a \pi_{\theta_o}(a|s) \log \pi_{\theta_o}(a|s) - \int_a \pi_{\theta_o}(a|s) \sum \lambda_i(s, a) \log \pi_i + \int_a \pi_{\theta_o}(a|s) \sum \lambda_i(s, a) (\log \pi_i - \frac{1}{\alpha} q_i(a|s))] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [\int_a \sum \lambda_i(s, a) (\pi_{\theta_o} \log \pi_{\theta_o} - \pi_{\theta_o} \log \pi_i) + \int_a \pi_{\theta_o}(a|s) \sum \lambda_i(s, a) (\log \frac{\pi_i}{\exp \frac{q_i}{\alpha}})] \\
&\leq \mathbb{E}_{s \sim \rho^{\pi_o}} [\int_a \sum \lambda_i(s, a) \pi_{\theta_o} \log \frac{\pi_{\theta_o}}{\pi_i} + \int_a \pi_{\theta_o}(a|s) \log \sum \lambda_i(s, a) \frac{\pi_i}{\exp \frac{q_i}{\alpha}}] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [\int_a \pi_{\theta_o} (\sum \lambda_i(s, a) \log \frac{\pi_{\theta_o}}{\pi_i} + \log \sum \lambda_i(s, a) \frac{\pi_i}{\exp \frac{q_i}{\alpha}})] \\
&= \mathbb{E}_{s \sim \rho^{\pi_o}} [\sum \lambda_i(s, \xi) D_{KL}(\pi_{\theta_o} || \pi_i) + \int_a \pi_{\theta_o} (\log \sum \lambda_i(s, a) \frac{\pi_i}{\exp \frac{q_i}{\alpha}})]
\end{aligned}$$

where $\lambda_i = \frac{|w_i|}{\sum |w_j|}$ $q_i = Q_i^{\pi_o} \frac{w_i \sum |w_j|}{|w_i|}$

(12)

Because the α is only used to make a constraint on the policy's entropy, we can use a vector of $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n, \alpha_o]^T$ to replace the scalar α . This operation will bring more randomness to the auxiliary policies, which may help finding the optimal policy. The loss function can be defined as :

$$\begin{aligned}
L_{\pi_o} &= \mathbb{E}_{s \sim D} \left[\sum \lambda_i(s, \xi) D_{KL}(\pi_{\theta_o} || \pi_i) + \int_a \pi_{\theta_o}(s, a) (\log \sum \lambda_i(s, a) \frac{\pi_i(s, a)}{\exp \frac{q_i}{\alpha_i}}) \right] \\
&= \mathbb{E}_{s \sim D} \left[\sum \lambda_i(s, \xi) D_{KL}(\pi_{\theta_o} || \pi_i) + \mathbb{E}_{a \sim \pi_o} \left[\log \sum \lambda_i(s, a) \frac{\pi_i}{\exp \frac{q_i}{\alpha_i}} \right] \right]
\end{aligned}$$

(13)

The determinacy of the policy π_o is gradually improved, hence we use the mean \bar{a} of Gaussian formed policy π to approximate ξ . The final version optimization objective of π_o can be formalized as follow:

$$\begin{aligned}
L_{\pi_o}(\theta_o) &= \mathbb{E}_{s \sim D} \left[\sum_i \lambda_i(s, \bar{a}) D_{KL}(\pi_{\theta_o} || \pi_i) + \mathbb{E}_{a \sim \pi_o} [\log \sum \lambda_i(s, a) \frac{\pi_i}{\exp \frac{q_i}{\alpha_i}}] \right] \\
&\text{where } i \in [1, 2, \dots, n]
\end{aligned}$$

(14)

B HYPERPARAMETERS

Table 1 lists the common MRF parameters used in the experiments mentioned in Table 1.

C SUPPLEMENTARY MATERIAL FOR EXPERIMENTS

C.1 RANDOM WALK

In this task, we exhibit the difference of Q_i 's mean and variance between using MRF with regularization and MRF without regularization in Fig.(9) and Fig.(10). We can find that, in this task, MRF without regularization cannot represent the state-action value function well. We consider this phenomenon results from the disabling of λ calculator. And this problem can be settled by the similarity-based regularization we proposed, as shown in section 4.1.

Table 1: MRF Hyperparameters

Parameter	Value
optimizer	Adam(Kingma & Ba (2014)) and PCGrad(Yang et al. (2019))
learning rate ($\beta_Q, \beta_{\pi_{aux}}, \beta_\varphi, \beta_{\pi_o}, \beta_\alpha$)	$3 \cdot 10^{-4}$
discount (γ)	0.99 (contains cell nucleus)
target smoothing coefficient(τ)	$5 \cdot 10^{-3}$
replay buffer size	10^6
number of hidden layers(all networks)	3
number of hidden units per layer	256
number of samples per minibatch	256
entropy target	$-\dim(A)$
nonlinearity	ReLU
target update interval	1
gradient steps	1

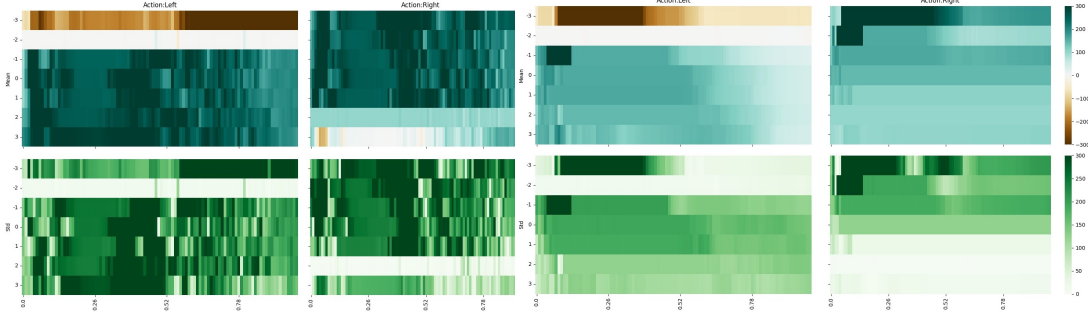


Figure 9: MRF without Regularization

Figure 10: MRF without Regularization

C.2 MOUNTAIN CAR

In this sparse reward task, we use two styles of the shaping rewards:

$$\begin{aligned}
 r_1 &= -\|S_G - s\|_2 \\
 r_2 &= \|S_l - s\|_2 \\
 \mathbf{r} &= r_o + [r_1, r_2, 0.0, 0.0]^T
 \end{aligned} \tag{15}$$

where \mathbf{r} is the vector of multi-perspective rewards, S_G is the goal state of Mountain Car task, S_l is the leftmost state of this task. Here, r_1 is used to tell the agent need to approach the goal state and r_2 is used to tell the agent keep away from the leftmost state in this task. These two shaping rewards correspond to the two different perspective of completing this task. It is obvious that our method can achieve better results than using scalar shaping reward, which means that via MRF the information of the shaping rewards can be better used.

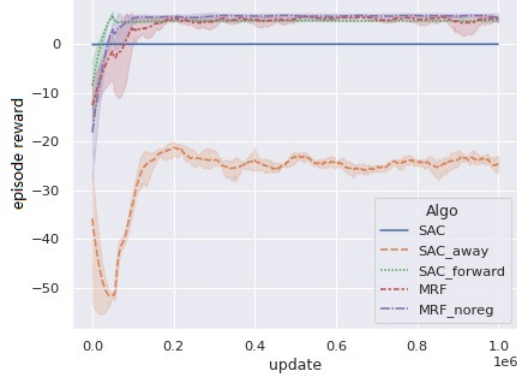


Figure 11: Performance in Mountain Car

C.3 LUNAR LANDER

We convert this task in Gym, previously a dense reward task, into a sparse reward task. Meanwhile, we decompose the previous version’s scalar shaping reward into multi-perspective rewards we need (only a linear decomposition). The scalar shaping reward of this task, in Gym, is:

$$r_{shaping} = -100 \cdot (\|P_G - P_s\|_2) - 100 \cdot (\|V_s\|_2) - 100 \cdot |Angle| + 10 \cdot \mathbb{1}_{left} + 10 \cdot \mathbb{1}_{right} \quad (16)$$

where P_G and P_s are the goal position and the agent position respectively, V_s is the agent’s velocity, $Angle$ means the angle between the agent’s body and the desired landing direction, $\mathbb{1}_{left}$ and $\mathbb{1}_{right}$ represent if the left or right leg contact the ground.

Through our method, we only need to care about the component with different information, and the multi-perspective rewards can be formalized as follow:

$$\mathbf{r} = r_o + [-\|P_G - P_s\|_2, -\|V_s\|_2, -|Angle|, \mathbb{1}_{left}, \mathbb{1}_{right}, 0.0, 0.0]^T \quad (17)$$

The performance of our method is not bad than SAC using selective shaping reward and the basic SAC.

C.4 HOPPER

Hopper is an environment with dense reward. The purpose of this kind of task is getting higher episode rewards. This environment is not easy to demonstrate the outstanding performance of reward shaping. We doing this experiment in this task is only to show the data efficiency of MRF and the advantage of multi-perspective rewards architecture. The original reward provided by the environment is:

$$r_o = R_{forward} + R_{healthy} - R_{energy} \quad (18)$$

We design the multi-perspective rewards as follow:

$$\mathbf{r} = r_o + [R_{forward}, R_{healthy}, -R_{energy}, 0.0, 0.0]^T \quad (19)$$

Here, we design the multi-perspective rewards by enforcing each component of the original shaping reward.