A Algorithms

Algorithm 1 Soft Actor Critic with Entropy Tuning

Initialize Q-functions, $Q_{\theta_1}(s, a)$, $Q_{\theta_2}(s, a)$ and policy weights $\pi_\phi(a_t|s_t)$
Initialize target networks $Q_{\bar{\theta_1}}(s, a)$, $Q_{\bar{\theta_2}}(s, a)$
Initialize replay buffer D
for each iteration **do**
  for each environment step **do**
    Sample action form policy π
    store transition into replay buffer
  end for
  for each gradient step **do**
    Update Q-function using $J_Q(\theta_i)$ for $i \in \{1, 2\}$ $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$
    Update policy weights using $J_\pi(\phi)$ $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    Adjust temperature using $J(\alpha)$ $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$
    Update target network weights using Polyak averaging $\bar{\theta_i} \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta_i}$
  end for
end for

Algorithm 2 Prioritized Experience Replay

Initialize replay buffer D, exponents α and β
for step t, ..., T **do**
  Choose action $a_t \sim \pi_\phi(s_t)$
  Observe s_{t+1}, r_{t+1}
  Store transition $(a_t, s_t, r_{t+1}, s_{t+1})$ in D
  assign maximal priority $p_t = max_{i<t} p_i$
  for sample k, ..., K **do**
    Sample transition k under equation 6
    Compute TD Error for samples using: $\delta_k = r_{k+1} + \gamma \max_{a \in A} Q_{\theta^-}(s_{k+1}, a) - Q_\theta(s_k, a_k)$
    $p(k) = \delta_k + \epsilon$
    $P(k) \leftarrow \frac{p(k)^\alpha}{\sum_i p(i)^\alpha}$
    $w_k = (\frac{1}{N} \cdot \frac{1}{P(k)})^\beta$
  end for
  take K gradient steps to minimize Bellman error weighted by w_k
end for

Algorithm 3 Emphasizing Recent Experience

Initialize Replay Buffer D, set $\eta_t = \eta_0$, episode length K = 0
for t=1,...,T **do**
    Choose action $a_t \sim \pi_\phi(s_t)$
    Observe s_{t+1}, r_{t+1}
    Store transition $(a_t, s_t, r_{t+1}, s_{t+1})$ in D
    $\eta_t \longleftarrow \eta_0 + (\eta_T - \eta_0) \cdot \frac{1}{T}$
    $t \longleftarrow t + 1$, $K \longleftarrow K + 1$
    if s_{t+1} is a terminal state **then**
        for step k in K mini-batch update **do**
            $c_k = max(N \cdot \eta^{k\frac{1000}{K}}, 5000)$
            $B \sim D_{c_k}$
            Perform Gradient step on B
        end for
        K = 0
    end if
end for

B Hyperparameters

Parameter	Value
SAC	
Optimizer	Adam (Kingma & Ba, 2017)
learning rate	$3 \cdot 10^{-4}$
discount(γ)	0.99
replay buffer size	10^6
entropy target	-dim(A)
nonlinearity	ReLU
target smoothing coefficient(τ)	0.005
target update interval	1
gradient steps	1
PER	
initial prioritized experience replay buffer exponents (α, β)	(0.5,0.4)
ERE	
initial recency emphasis coefficient	0.996
terminal recency emphasis coefficient	1.0

Table 2: SAC, PER, and ERE hyperparameters

C Experiments

C.1 Experiments on baseline performance in continuous control

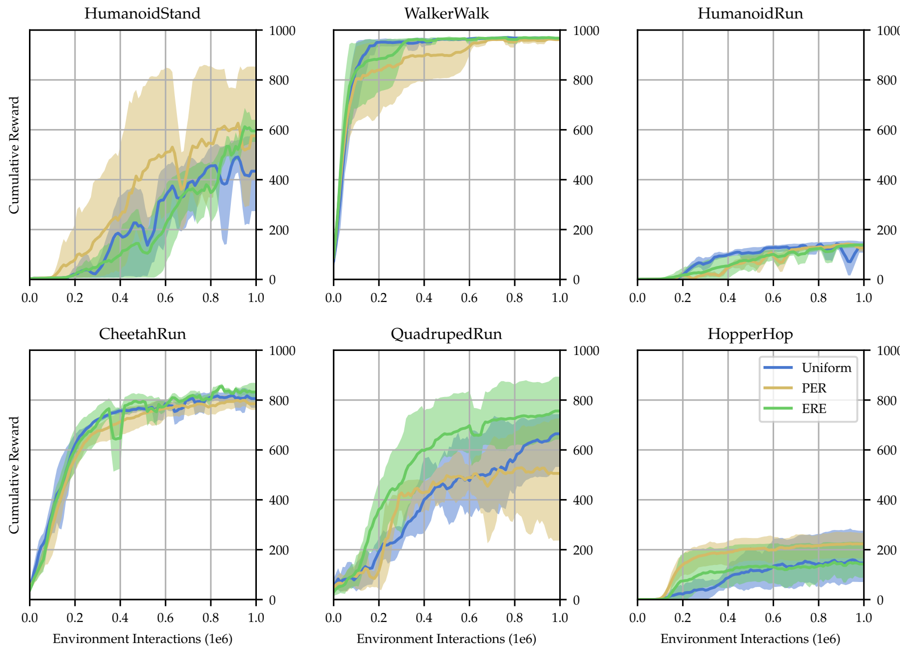
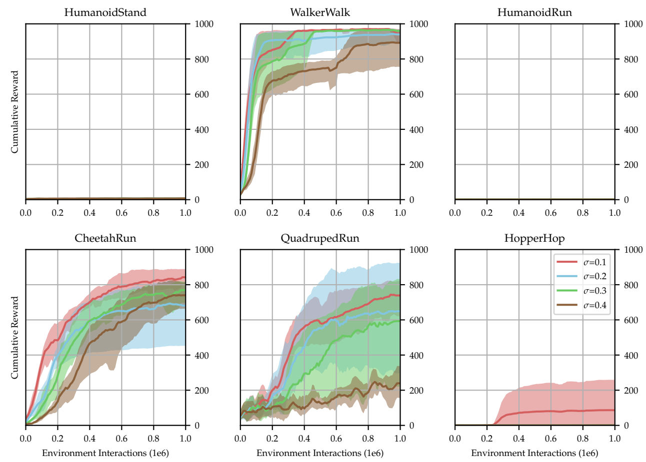
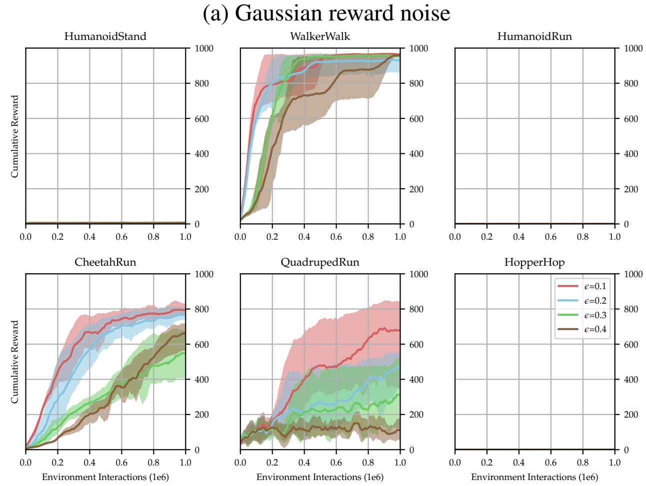


Figure 8: Detailed learning curves of Uniform (blue), PER (yellow), and ERE (green) on 6 Deepmind Control tasks. The solid lines are the median scores while the shaded area denotes the interquantile range across 5 random seeds

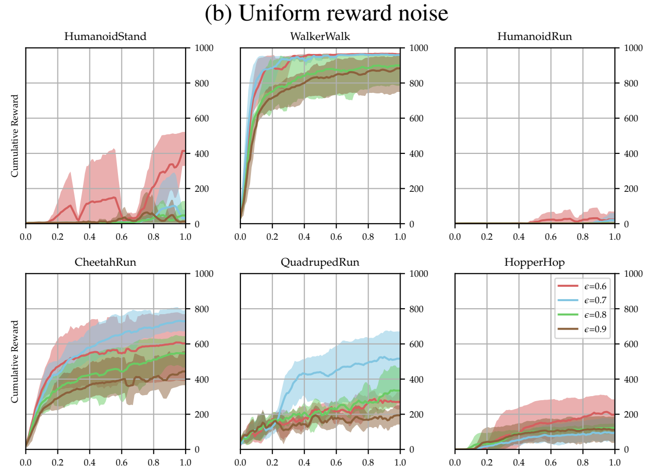
C.2 Experiments on sampling methods with added reward noise



(a) Gaussian reward noise

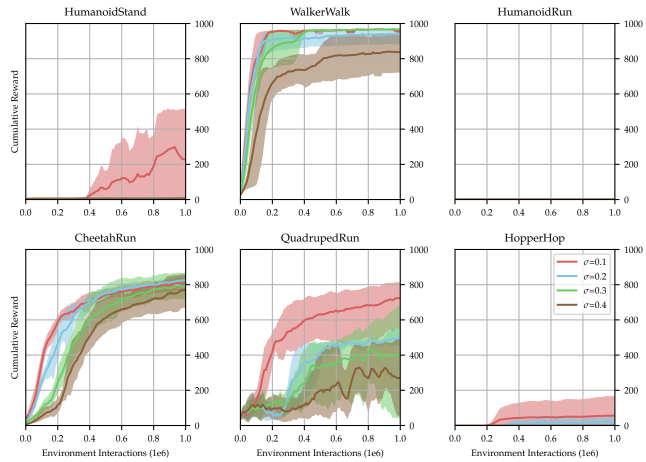


(b) Uniform reward noise

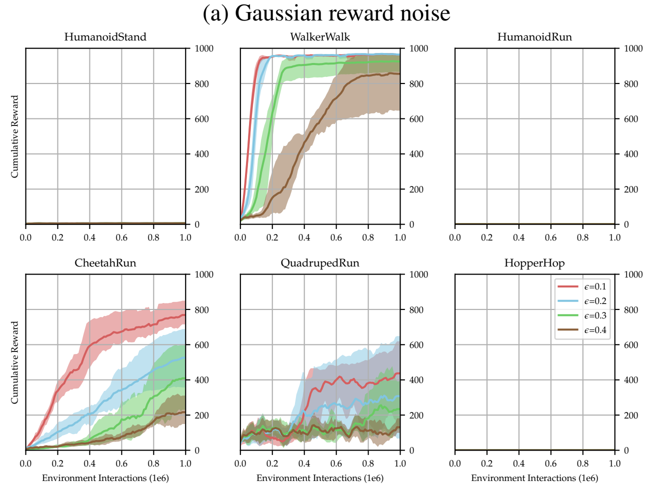


(c) Sparse reward noise

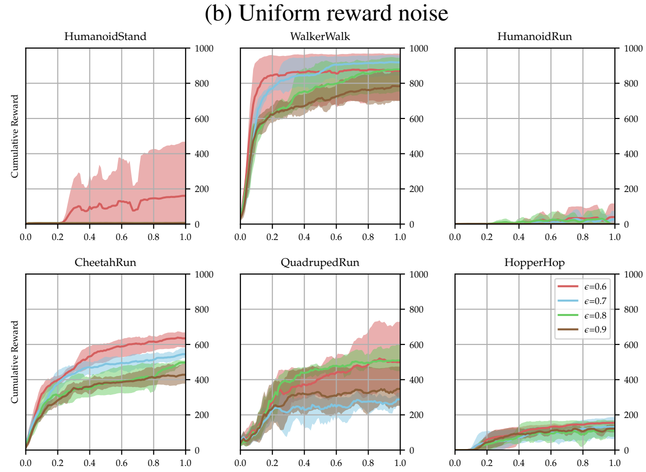
Figure 9: Learning curves for a) Gaussian, b) Uniform, and c) Sparse added reward noise under Uniform sampling. The shaded area corresponds to the interquantile range across 5 random seeds. In some environment the addition of noise results to catastrophic failure leading to close to 0 cumulative reward



(a) Gaussian reward noise

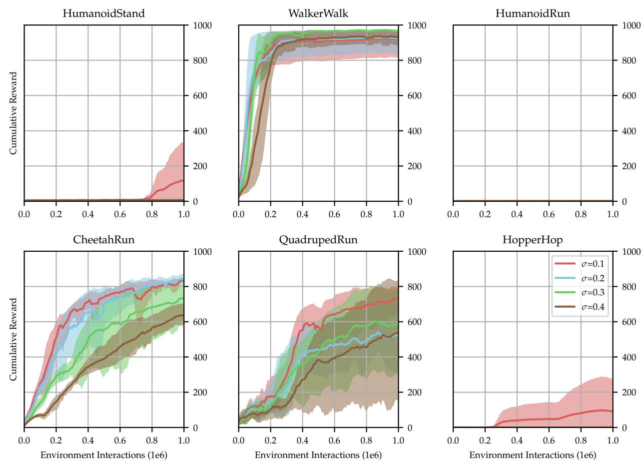


(b) Uniform reward noise

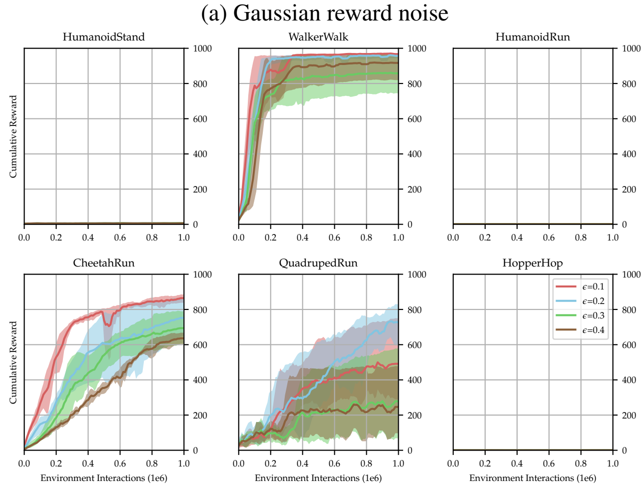


(c) Sparse reward noise

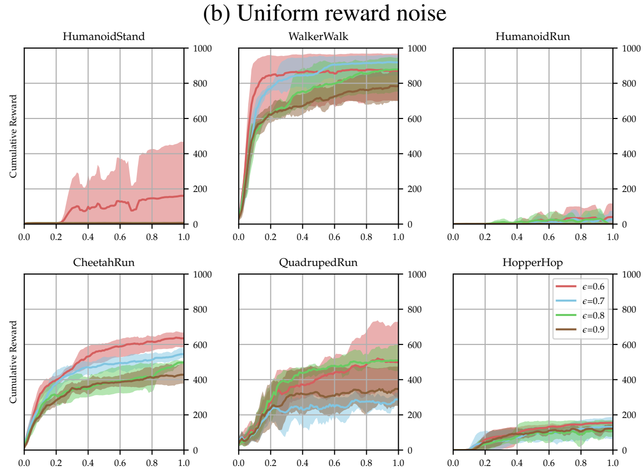
Figure 10: Learning curves for a) Gaussian, b) Uniform, and c) Sparse added reward noise with PER



(a) Gaussian reward noise



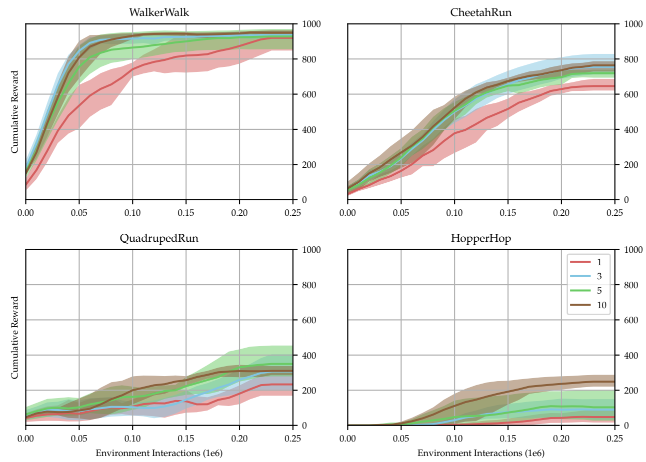
(b) Uniform reward noise



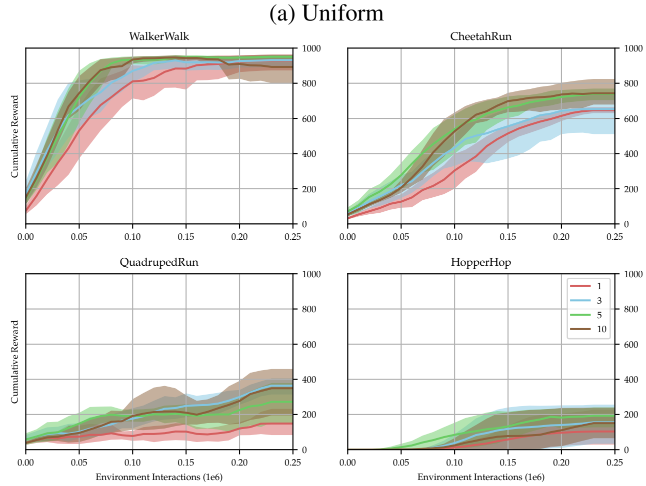
(c) Sparse reward noise

Figure 11: Learning curves for a) Gaussian, b) Uniform, and c) Sparse added reward noise with ERE

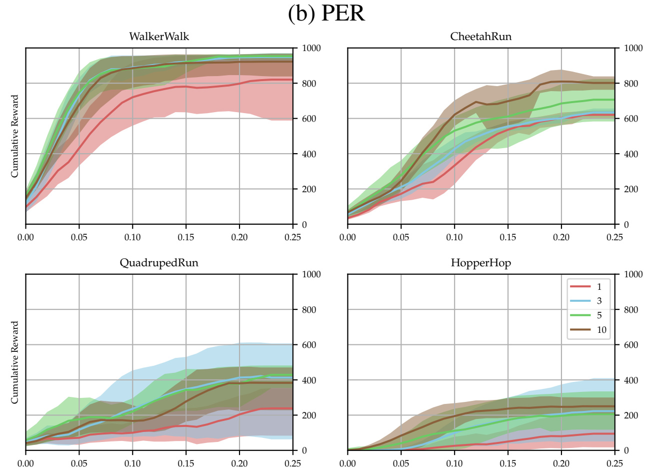
C.3 Experiments on replay buffer sensitivity analysis



(a) Uniform

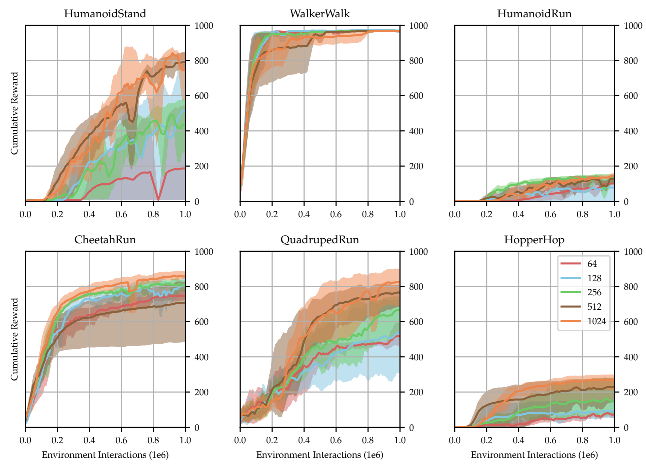


(b) PER

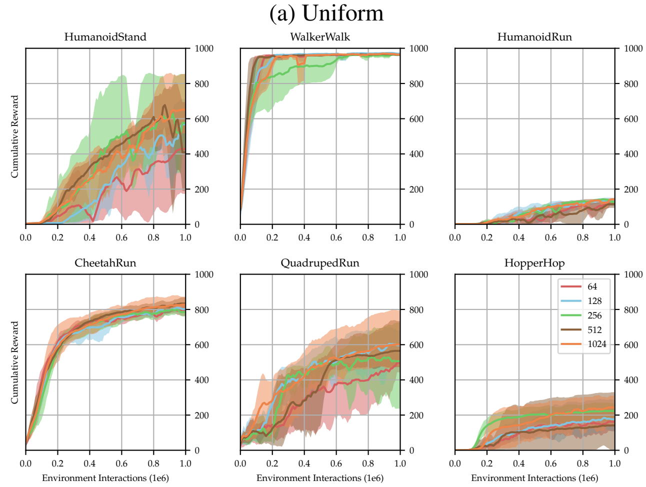


(c) ERE

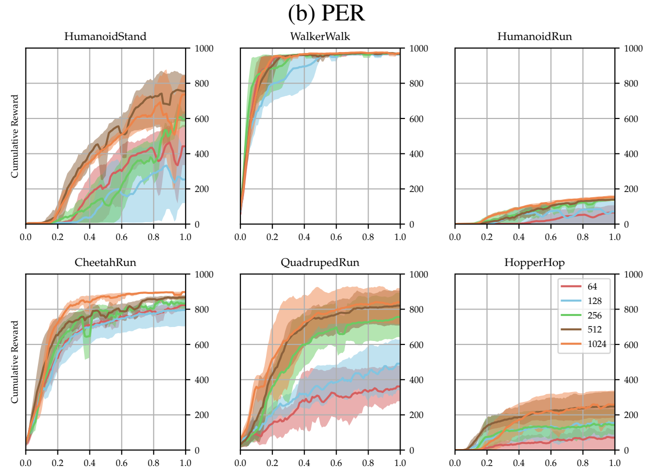
Figure 12: Learning curves for varying number of policy updates over $0.25 \cdot 10^6$ environment interactions for a) Uniform sampling, b) PER, c) ERE.The shaded area corresponds to the interquantile range across 5 random seeds



(a) Uniform

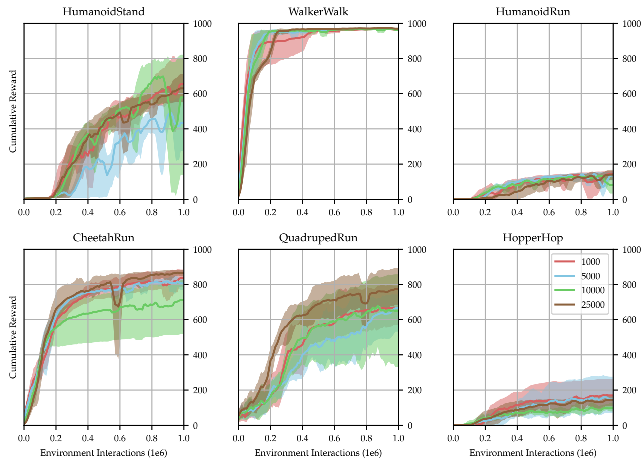


(b) PER

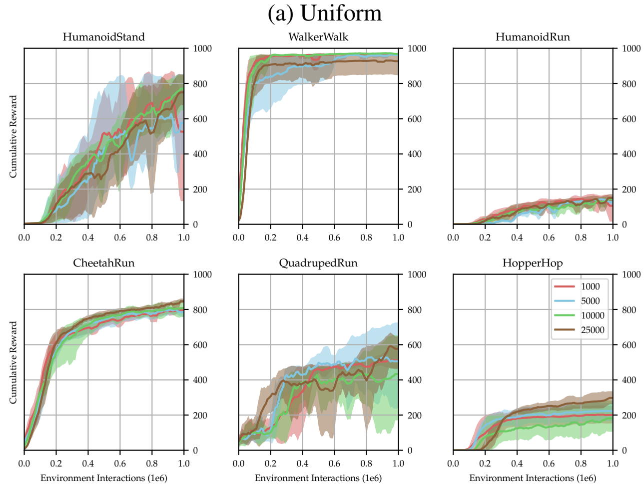


(c) ERE

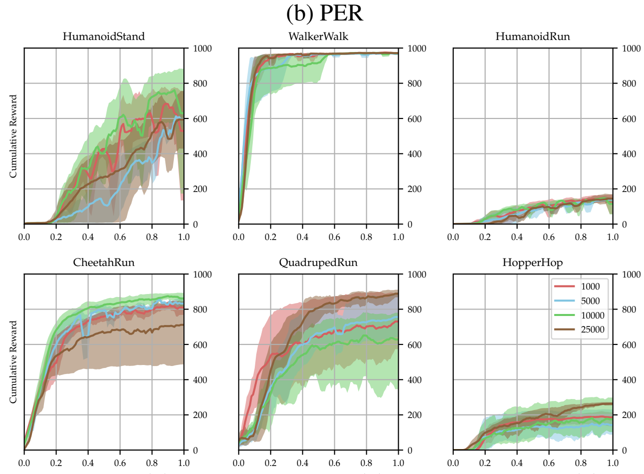
Figure 13: Learning curves for varying batch size for a) Uniform sampling, b) PER, c) ERE



(a) Uniform

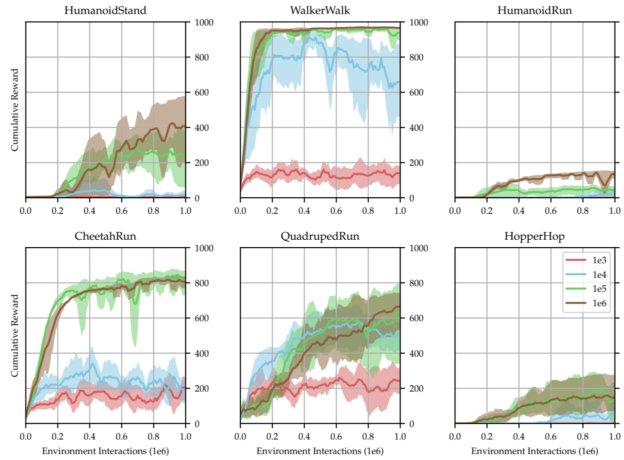


(b) PER

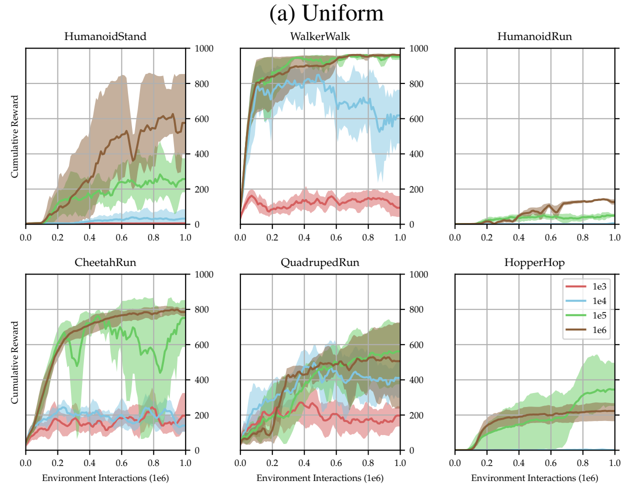


(c) ERE

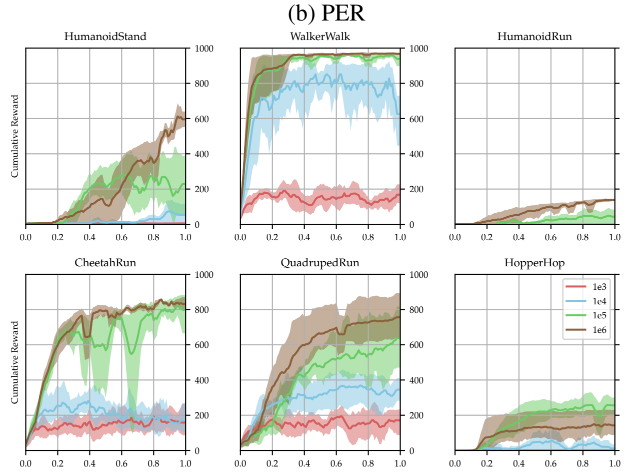
Figure 14: Learning curves for varying exploration steps for a) Uniform sampling, b) PER, c) ERE



(a) Uniform



(b) PER



(c) ERE

Figure 15: Learning curves for varying replay buffer capacity for a) Uniform sampling, b) PER, c) ERE.

C.4 Experiments on separate sampling merhods per Actor-Critic and varying priority metrics

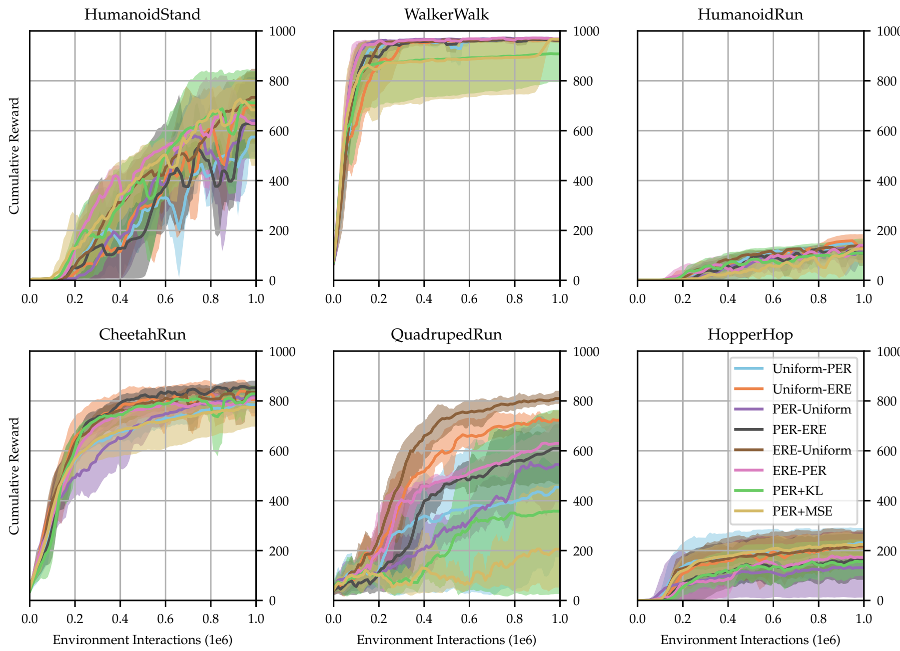


Figure 16: Learning curves of separate sampling methods per Actor-Critic and PER with different prioritization metrics(MSE, KL divergence). The shaded area corresponds to the interquantile range across 5 random seeds