COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

Data Analytics at Texas A&M (DATA Lab)

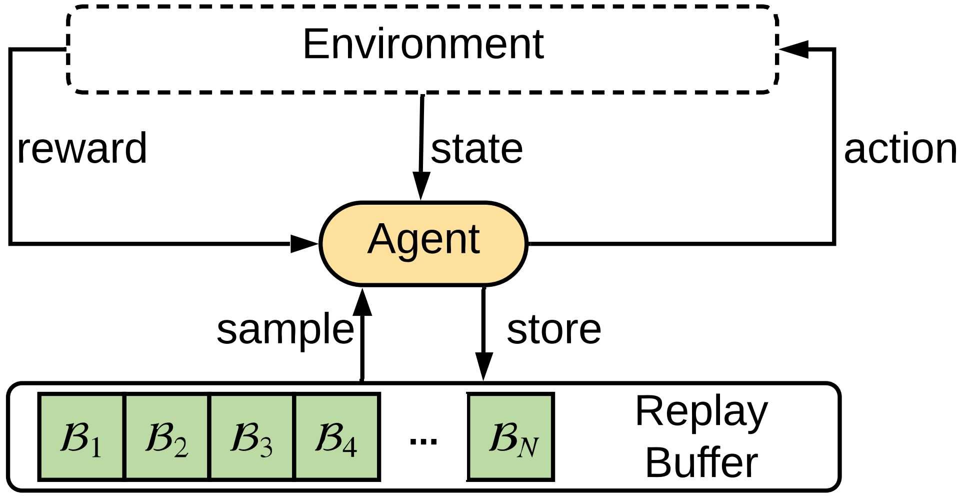
Experience Replay Optimization

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou and Xia Hu

Department of Computer Science and Engineering, Texas A&M University

Emails: {daochen.zha, khlai037, zkxiong, xiahu}@tamu.edu

What is Experience Replay



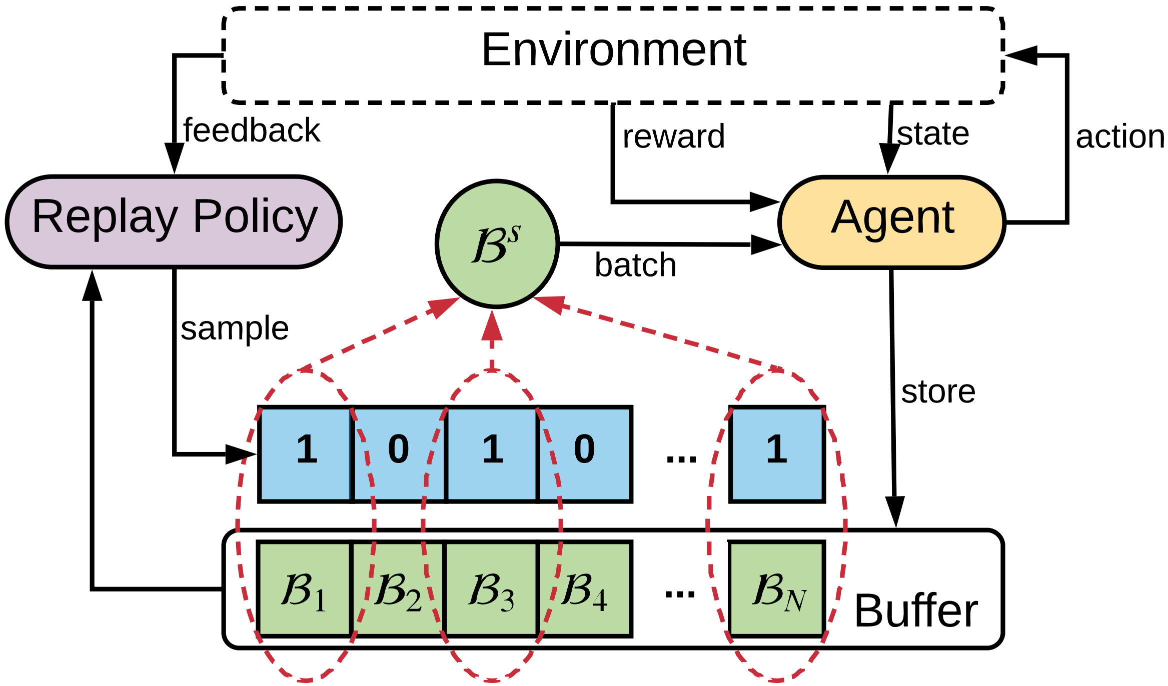
- ***Definition:*** *A memory buffer that stores past transitions (experiences) which are replayed for later use.*
- *A key technique behind contemporary off-policy RL algorithms. It greatly stabilizes the training and improves the sample efficiency.*

Replay Strategy Matters

- ***Uniform Sampling:*** *Sample transitions in the memory with equal probabilities. Applied in most off-policy algorithms.*
- ***Prioritized Experience Replay [1]:*** *Prioritize the transitions with higher temporal differences (TD) errors. Improved performance for DQN on Atari environments.*
- ***Various Memory Size [2][3]:*** *Manage the size of the replay buffer.*
- ***Remember/Forget Experience [4]:*** *Selectively remember/forget experience can improve the performance.*

[1] Marcelo Gomes Mattar and Nathaniel D Daw. "Prioritized experience replay." ICLR 2016.
[2] Zhang and Sutton. "A deeper look at experience replay" NIPS Deep Reinforcement Learning Symposium, 2017.
[3] Liu and Zou. "The effects of memory replay in reinforcement learning". Arxiv 2017.
[4] Novati and Koumoutsakos. "Remember and forget for expe- rience replay". ICML 2019.

Experience Replay Optimization (ERO)



- *Humans tend to replay the memories that will lead to the most rewarding future decisions.*
- *We are motivated to use the feedback from the environment as a rewarding signal to adjust the replay strategy.*

Sampling with Replay Policy

- *A priority score function* $\phi(\mathbf{f}_{\mathcal{B}_i}|\theta^{\phi}) \in (0,1)$ where $\mathbf{f}_{\mathcal{B}_i}$ are some features for a transition.
- Given the scores for all the transitions

$$\boldsymbol{\lambda} = \left\{\phi\left(\mathbf{f}_{\mathcal{B}_i}\middle|\theta^{\phi}\right)\middle|\mathcal{B}_i \in \mathcal{B}\right\}.$$

- *Before feeding into standard RL training, we narrow down all the transitions to a subset of transitions* $\mathcal{B}^s$*:*

$$\mathbf{I}\sim\mathrm{Bernoulli}(\boldsymbol{\lambda}),$$

$$\mathcal{B}^s = \{\mathcal{B}_i|\mathcal{B}_i \in \mathcal{B} \wedge \mathbf{I}_i = 1\}.$$

- *Then* $\mathcal{B}^s$ *is used to update the agent with standard procedures.*

Training with Policy Gradient

- The replay-reward is defined as the improvement of the cumulative reward:

$$r^r = r^c_{\pi} - r^c_{\pi'}.$$

where r^c_{π} and $r^c_{\pi'}$ are the recent cumulative rewards.

- By using the REINFORCE trick, we can calculate the gradient of the improvement $\mathcal{J}$ w.r.t θ^{ϕ}:

$$\begin{aligned} \nabla_{\theta^{\phi}} \mathcal{J} &= \nabla_{\theta^{\phi}} \mathbb{E}_{\mathbf{I}}[r^r] \\ &= \mathbb{E}_{\mathbf{I}}\left[r^r \nabla_{\theta^{\phi}} \log \mathrm{P}(\mathbf{I}|\phi)\right]. \end{aligned}$$

- *Where* ϕ is the abbreviation for $\phi(f_{\mathcal{B}_i}|\theta^{\phi})$.

Applying to Off-Policy Algorithms

Algorithm 1 ERO enhanced DDPG

DDPG

Initialize policy π, replay policy ϕ, buffer $\mathcal{B}$
for each iteration **do**
  for each timestep t **do**
    Select action a_t according to π and state s_t
    Execute action a_t and observe s_{t+1} and r_t
    Store transition (s_t, a_t, r_t, s_{t+1}) into $\mathcal{B}$
    if episode is done **then**
      Calculate the cumulative reward r_π^c
      if $r_{\pi'}^c \neq$ null **then**
        $\mathcal{B}^s$ = UpdateReplayPolicy(r_π^c, $r_{\pi'}^c$, $\mathcal{B}$)
      end if
      Set $r_{\pi'}^c \leftarrow r_\pi^c$
    end if
  end for

DDPG

  for each training step **do**
    Uniformly sample a batch $\{\mathcal{B}_i^s\}$ from $\mathcal{B}^s$
    Update the critic of π with Eq. (9) and (10)
    Update the actor of π with Eq. (11)
    Update target networks with Eq. (12) and (13)
    Update λ for each transition in $\{\mathcal{B}_i\}$
  end for
end for

Algorithm 2 UpdateReplayPolicy

Input:
Cumulative reward of current policy r_π^c
Cumulative reward of previous policy $r_{\pi'}^c$
Buffer $\mathcal{B}$
Output:
Sampled subset $\mathcal{B}^s$

Calculate replay-reward based on Eq. (3)
for each replay updating step **do**
  Randomly sample a batch $\{\mathcal{B}_i\}$ from $\mathcal{B}$
  Update replay policy based on Eq. (8)
end for
Sample a subset $\mathcal{B}^s$ from $\mathcal{B}$ using Eq. (2)

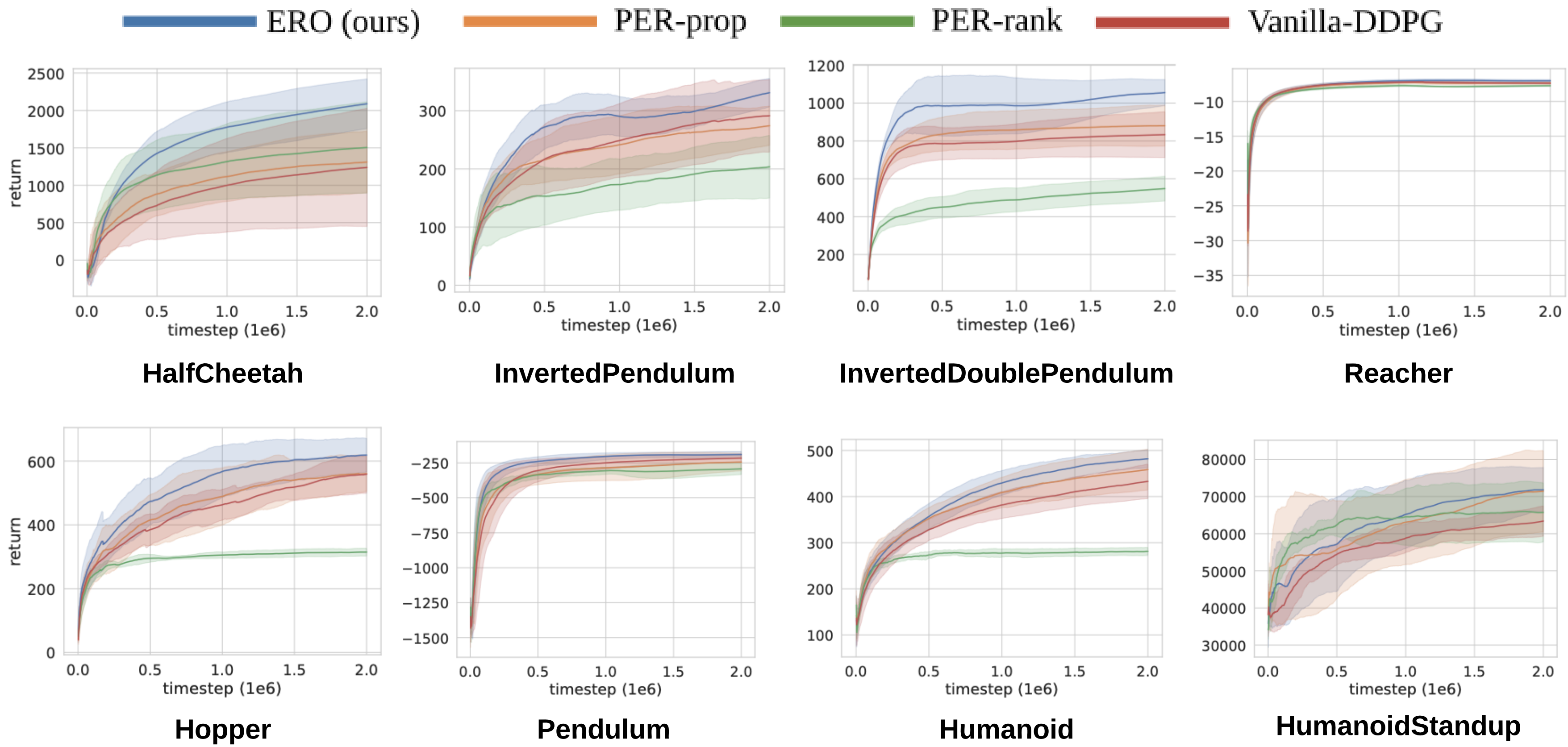
Update Replay Policy

Sample New Transitions

Experimental Settings

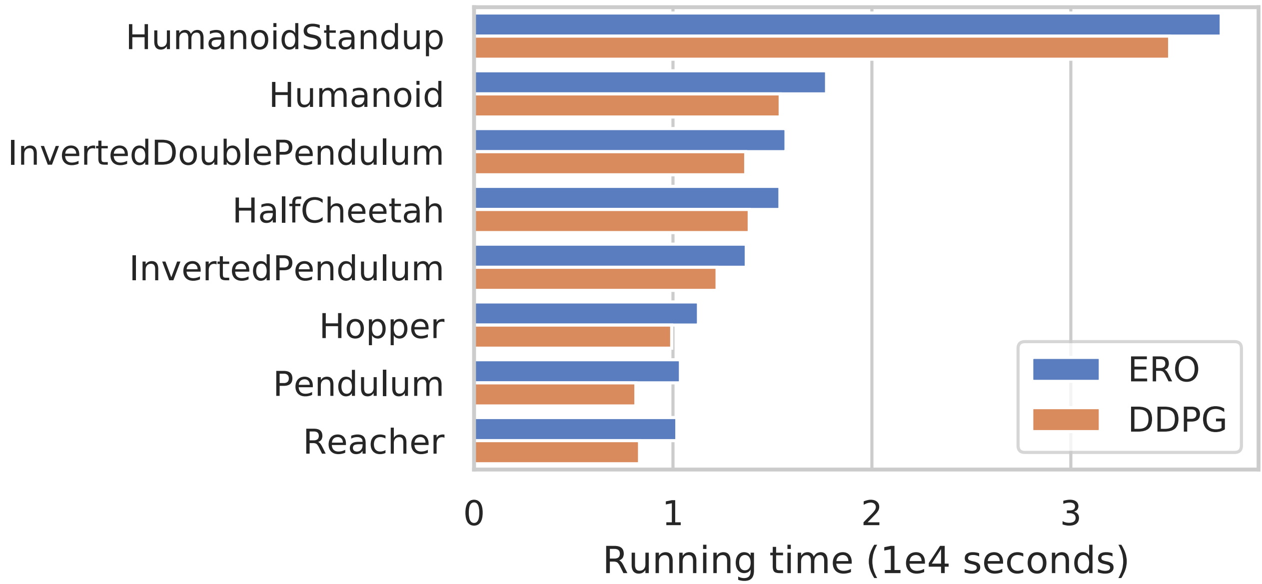
- ***Baselines:*** *Vanilla-DDPG, Proportional Prioritized Experience Replay (PER-pro), Rank-based PER (PER-rank).*
- ***Environments:*** *Continuous control tasks from OpenAI gym.*
- ***Implementation Details:***
    - *DDPG implementation in OpenAI baselines.*
    - *For ERO, three features are used: the reward of the transition, the temporal difference (TD) error, and the current timestep.*
    - *The replay policy is MLP with two hidden layers (64-64).*

Effectiveness Evaluation



- *ERO outperforms Vanilla-DDPG and rule-based replay strategy (PER-prop and PER-rank).*

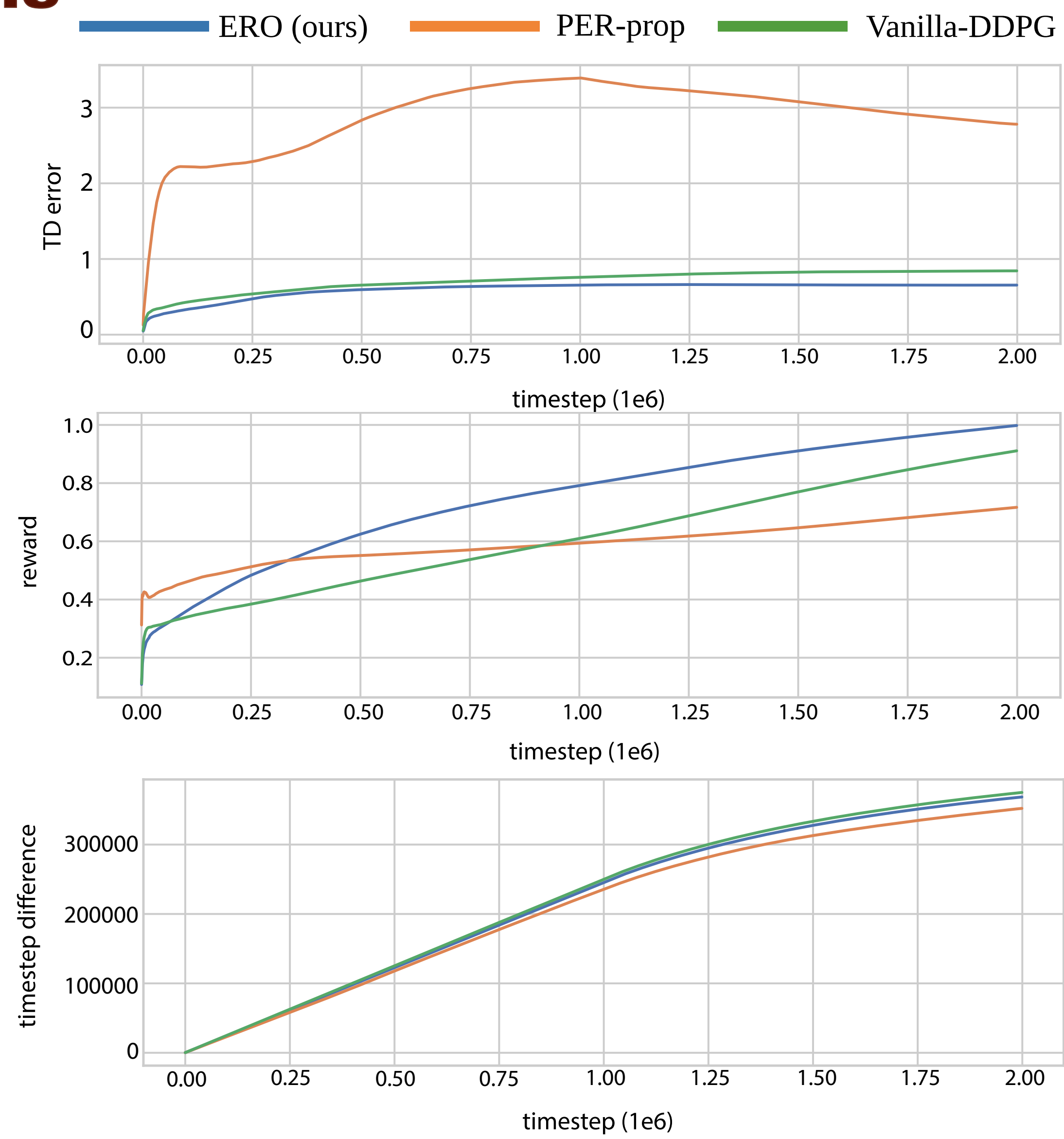
Efficiency Evaluation



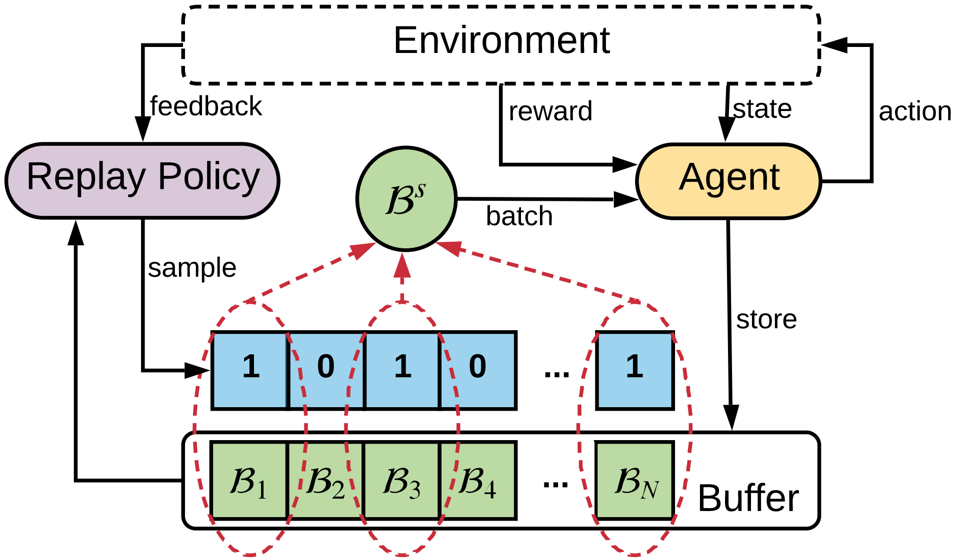
- *ERO only requires slightly more computation than Vanilla-DDPG for replay policy update.*

Replay Policy Analysis

- *The learned replay policy of ERO samples more transitions with low TD errors in HalfCheetah (More studies are needed to understand this aspect in the future work).*
- *ERO samples more recent transitions than Vanilla-DDPG.*



Conclusions



- *Formulate experience replay as a learning problem.*
- *Propose ERO, a general framework for effective and efficient use of replay memory.*
- *Conducted experiments on 8 continuous control tasks from OpenAI Gym demonstrate the effectiveness of ERO.*

Acknowledgement

- *DATA Lab and collaborators*

COMPUTER SCIENCE
& ENGINEERING
TEXAS A&M UNIVERSITY

DATA LAB
TAMU

Data Analytics at Texas A&M (DATA Lab)

- *National Science Foundation (NSF)*
- *Everyone attending the talk*